Similarity Measures for Semantic Relation Extraction

The dissertation is presented by Alexander Panchenko in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Jury : Prof. Cédrick Fairon (supervisor), Université catholique de Louvain
Prof. Andrey Philippovich (supervisor), Bauman Moscow State Technical University
Prof. Henri Bouillon (jury president) Université catholique de Louvain
Prof. Marco Saerens, Université catholique de Louvain
Dr. Jean-Michel Renders, Xerox Research Center Europe
Prof. Marie-Francine Moens, KU Leuven

Louvain-la-Neuve
2012-2013

To my parents Luidmila and Ivan
for their unconditional love and support.

Contents

Acknowledgments

First of all, I thank my supervisor professor Cédrick Fairon from Université catholique de Louvain and co-supervisor professor Andrey Philippovich from Bauman Moscow State Technical University for their countless help and support during these years. Next, I would like to acknowledge financial support of "Wallonie-Bruxelles International (WBI)" foundation and "Institut Langage et Communication (IL&C)" of Université catholique de Louvain. I am also thankful to the members of my scientific committee: professor Marco Saerens from Université catholique de Louvain, Dr. Jean-Michel Renders from Xerox Research Center and professor Marie-Francine Moens from KU Leuven. Their advanced questions, precise suggestions and critical comments significantly improved quality of this dissertation. Moreover, I want to thank professor Yuri N. Philippovich from Bauman State Technical University for helping me make the first steps in Computational Linguistics and for all our scientific discussions.

CENTAL, the NLP laboratory of Université catholique de Louvain, provided me an excellent research environment. I especially acknowledge help of Dr. Thomas François, who was always ready to answer a question and share his knowledge of Statistics and Natural Language Processing. Several people from CENTAL provided helpful comments on the first versions of this text: Adrien Dessy, Olga Morozova, Jean-Léon Bouraoui, Thomas François, Sandrine Brognaux, Hubert Naets, Stéphanie Weiser, Patrick Watrin, Adrien Bibal and Louise-Amélie Cougnon. This help was essential to the success of the work.

Last but not least, I thank all contributors to the "Serelex" project, especially Pavel Romanov, Hubert Naets, Olga Morozova and Alexey Romanov. It was a great pleasure and fun to collaborate with you.

Finally, I thank Polina for love and support.

Alexander Panchenko

Louvain-la-Neuve, 14th February 2013

Publications Related to this Thesis

[1] *Panchenko A., Beaufort R., Naets H., Fairon C.* Towards Detection of Child Sexual Abuse Media: Classification of the Associated Filenames. In Proceedings of the 35th European Conference on Information Retrieval (ECIR 2013). Lecture Notes in Computer Science (Springler), vol.7814, Moscow, Russia.

[2] *Panchenko A., Romanov P., Morozova O., Naets H., Philippovich A., Romanov A., Fairon C.* Serelex: Search and Visualization of Semantically Related Words. In Proceedings of the 35th European Conference on Information Retrieval (ECIR 2013). Lecture Notes in Computer Science (Springler), vol.7814, Moscow, Russia.

[3] *Panchenko A., Morozova O., Naets H.* A Semantic Similarity Measure Based on Lexico-Syntactic Patterns. // In Proceedings of the 11th Conference on Natural Language Processing (KONVENS 2012), — Vienna (Austria), 2012 – pp.174–178.

[4] *Panchenko A., Beaufort R., Fairon C.* Detection of Child Sexual Abuse Media on P2P Networks: Normalization and Classification of Associated Filenames. // In Proceedings of Public Security Applications Workshop, International Conference on Language Resources and Evaluation (LREC 2012) — Istanbul (Turkey), 2012 – pp. 27-31.

[5] *Panchenko A., Adeykin S., Romanov P., Romanov A.* Extraction of Semantic Relations between Concepts with KNN Algorithms on Wikipedia. // In Proceedings of Concept Discovery in Unstructured Data Workshop (CDUD), International Conference On Formal Concept Analysis (ICFCA 2012) — Leuven (Belgium), 2012.

[6] *Panchenko A., Morozova O.* A Study of Hybrid Similarity Measures for Semantic Relation Extraction. // In Proceedings of Innovative Hybrid Approaches to the Processing of Textual Data Workshop, Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012) — Avignon (France), 2012 — pp. 10–18.

[7] *Panchenko A.* A Study of Heterogeneous Similarity Measures for Semantic Relation Extraction. // In Proceedings of 14e Rencontres des Étudiants Chercheurs en Infor-

matique pour le Traitement Automatique des Langues (JEP-TALN-RECITAL 2012) — Grenoble (France), 2012 — pp. 29–42.

[8] *Panchenko A.* Towards an Efficient Combination of Similarity Measures for Semantic Relation Extraction. // Abstract in Computational Linguistics in the Netherlands (CLIN 22) – Tilburg (The Netherlands): Tilburg University, 2012 – pp.6.

[9] *Panchenko A.* Comparison of the Knowledge-, Corpus-, and Web-based Similarity Measures for Semantic Relations Extraction. // In Proceedings of GEometrical Models of Natural Language Semantics Workshop (GEMS), Conference on Empirical Methods in Natural Language Processing (EMNLP 2011) – Edinburgh (UK), 2011 – pp. 11–21.

[10] *Panchenko A.* Comparison of the Knowledge-, Corpus-, and Web-based Similarity Measures for Semantic Relations Extraction. // Poster in Russian Young Scientists Conference in Information Retrieval (YSC 2011), Russian Summer School in Information Retrieval (RuSSIR 2011) — Saint-Petersburg (Russia), 2011.

[11] *Panchenko A.* Can We Automatically Reproduce Semantic Relations of an Information Retrieval Thesaurus? // In Proceedings of Russian Young Scientists Conference in Information Retrieval (YSC 2010), Russian Summer School in Information Retrieval (RuSSIR 2010) — Voronezh (Russia), 2010. – pp. 36–51.

`http://elar.usu.ru/bitstream/1234.56789/3058/1/russir-2010-04.pdf`

[12] *Panchenko A.* Computing Semantic Relations from Heterogeneous Evidence. // Abstract in Computational Linguistics in the Netherlands (CLIN 21) – Ghent (Belgium): University College Ghent, 2011 – pp. 39.

In Russian

[13] *Панченко А., Адейкин С., Романов П., Романов А.* Извлечение семантических отношений из статей Википедии с помощью алгоритмов ближайших соседей. // Труды конференции Анализ Социальных Сетей, Изображений и Текстов (АИСТ) — Екатеринбург, 2012 — С. 208–219.

[14] *Панченко А.* Метод автоматического построения семантических отношений между концептами информационно-поискового тезауруса. // Вестник Воронежского Государственного Университета. Серия “Системный Анализ и Информационные Технологии”, 2010 — Том 2. — С. 160–168.

http://www.vestnik.vsu.ru/program/view/view.asp?sec=analiz&year=2010&num=02&f_name=2010-02-26

[15] *Панченко А.* Towards an Efficient Combination of Similarity Measures for Semantic Relations Extraction. // Тезисы докладов научно-технической международной молодежной конференции "Системы, методы, техника и технологии обработки медиаконтента" — МГУП, 2011. — С. 3–4.

[16] *Панченко А.* Технология построения информационно-поискового тезауруса. // Интеллектуальные технологии и системы. Сборник учебно-методических работ и статей аспирантов и студентов, Издательство "Эликс+", 2009. — Том. 10. — С.124–140.

[17] *Панченко А.* Построение тезауруса из корпуса текстов предметной области. // Тезисы докладов "Жизнь языка в культуре и социуме-2" — Москва: Институт языкознания РАН, 2009.

[18] *Панченко А.* Методы автоматического извлечения семантических отношений. // Тезисы докладов "Электронные средства информации в современном обществе" — Москва: МГУП, 2010. — С. 59–61.

List of Notations and Abbreviations

C	a set of terms
$c_i \in C$	a term
R	a set of semantic relations
$\langle c_i, c_j \rangle \in R$	an untyped semantic relation
T	semantic relation type
syn	synonym
$hyper$	hypernym
$hypo$	hyponym
$cohypo$	co-hyponym
$mero$	meronym
$event$	object-event relation, i. e. an event that may occur with the object
$attri$	object-attribute relation
$random$	random relation, i. e. a pair of semantically unrelevant terms
$\langle c_i, t, c_j \rangle \in R$	a typed semantic relation
(C, R)	a semantic resource
$\hat{R}$	a set of extracted semantic relations
$\hat{R}_t \subseteq \hat{R}$	a subset of extracted semantic relations of type t
$\mathbf{F}$	a feature matrix
$\mathbf{f}_i$	a feature vector representing the term c_i

f_{ij}	the j-th feature representing the term c_i
$\mathbf{S}$	a $\mid C \mid \times \mid C \mid$ term-term similarity matrix
$s_{ij} \in \mathbf{S}$	a similarity score between terms c_i and c_j
d_{ij}	a dissimilarity score between terms c_i and c_j
$\cos(\mathbf{f}_i, \mathbf{f}_j)$	cosine between two feature vectors
$\parallel \mathbf{f} \parallel$	Euclidean (L2) norm of a feature vector
$\parallel \mathbf{f} \parallel_1$	Manhattan (L1) norm of a feature vector
$\parallel \mathbf{f} \parallel_\infty$	Maximum norm of a feature vector
$top(k, c_i)$	a set of k most similar terms to the term c_i
$percentile(p)$	p-th percentile of the similarity matrix $\mathbf{S}$
r	Pearson correlation coefficient
ρ	Spearman's rank correlation coefficient
$Precision(k)$ or $P(k)$	precision at $k\%$ on the semantic relation ranking task (Section 1.2.3)
$Recall(k)$ or $R(k)$	recall at $k\%$ on the semantic relation ranking task (Section 1.2.3)
$Fmeasure(k)$ or $F(k)$	F1-measure at $k\%$ on the semantic relation ranking task (Section 1.2.3)
$Precision$ at k	precision of the first k relations on the semantic relation extraction task (Section 1.2.3)
$Precision^E$	exact precision on the semantic relation extraction task (Section 2.2.3)
$Precision^{Fk}$	fuzzy precision on the semantic relation extraction extraction task, where k is the maximum shortest path between terms (Section 2.2.3)
B	a set of syntactic contexts
$\beta \in B$	a syntactic context
$P(c_i)$	probability of the i-th term
$P(c_i, f_j)$	probability that the i-th term is represented with the j-th feature
$Percent$	percent of the semantic relations of type t among all correctly extracted relations on the semantic relation ranking task (Section 3.6.3)

$Percent(k)$	percent of relations of type t among all correctly extracted relations at $k\%$ on the semantic relation ranking task (Section 3.6.3)
x_{ij}	dissimilarity of two semantic similarity measures sim_i and sim_j calculated with the χ^2-statistic (3.20)
sim_k	k-th semantic similarity measure
sim_{cmb}	hybrid semantic similarity measure
cmb	combination method
$\mathbf{S}_k$	a $\mid C \mid \times \mid C \mid$ term-term similarity matrix generated with the similarity measure sim_k
$s_{ij}^k \in \mathbf{S}_k$	a similarity score between terms c_i and c_j calculated with the similarity measure sim_k
$\mathbf{S}_{cmb}$	a $\mid C \mid \times \mid C \mid$ term-term similarity matrix generated with the hybrid similarity measure
$s_{ij}^{cmb} \in \mathbf{S}_{cmb}$	a similarity score between terms c_i and c_j calculated with the hybrid similarity measure sim_{cmb}
$\mathbf{x}$	a vector of similarity scores $(s_{ij}^1, \ldots, s_{ij}^K)$ that represents a pair of terms $\langle c_i, c_j \rangle$
w_k	a weight of the similarity measure sim_k
$\mathbf{w}$	a weight vector $(w_1, \ldots, w_K)$
z	a weighted linear combination of similarity measures (4.29)
$Accuracy$	classification accuracy
MC	human judgements dataset of Miller and Charles (1991) (Section 1.2.3)
RG	human judgements dataset of Rubenstein and Goodenough (1965) (Section 1.2.3)
WordSim	human judgements dataset WordSim353 (Section 1.2.3)
BLESS	Baroni and Lenci Evaluation of Semantic Spaces dataset (Section 1.2.3)
SN	Semantic Neighbors dataset (Section 1.2.3)
WordNet	Princeton WordNet 3.0 lexical database (Section 1.1.2)

measure	a semantic similarity measure, if not stated otherwise
corpus-based measure	a measure that derives similarity scores from a text corpus (Section 3.3)
web-based measure	a measure that derives similarity scores using the Web as a corpus approach (Section 3.3.2)
definition-based measure	a measure that derives similarity scores from a set of explicit term definitions (Section 3.4)
network-based measure	a measure that relies on the structure of a semantic network to derive the similarity scores (Section 3.2)
single measure	a measure that relies on one resource to derive the similarity scores, such as a semantic network (Chapters 2, 3)
hybrid measure	a measure that relies on several resources to derive the similarity scores (Chapter 4)
precision	precision on the semantic relation ranking task or precision on the semantic relation extraction task, depending on the context
LSA	Latent Semantic Analysis (Section 3.3.3)
k-NN	k nearest neighbor procedure (Section 1.2.2)
mk-NN	mutual k nearest neighbors procedure
POS	part of speech, e. g. “noun”,“verb” or “adjective”
FST	Finite State Transducer
pattern	lexico-syntactic pattern for extraction of relations from text
RelationFusion	a combination method (4.9)
WaCky	a dependency-parsed corpus of Wikipedia articles (Baroni et al., 2009)
ukWaC	a corpus of Web pages (Baroni et al., 2009)
PukWaC	a dependency-parsed corpus of Web pages (Baroni et al., 2009)
SDA	a corpus-based measure based on the Syntactic Distributional Analysis (Section 3.3.1)

SDA-k-cos	a corpus-based measure based on the Syntactic Distributional Analysis using k syntactic dependencies and the cosine (Section 3.3.1)
SDA-MWE	a corpus-based measure that relies on the Syntactic Distributional Analysis and supports multiword expressions (Section 2.2.2)
PatternSim	a corpus-based measure that relies on patterns (2.14-2.20)
PatternSim-EfreqRnum	the corpus-based measure *PatternSim* that uses the ranking formula *Efreq-Rnum* (2.17)
Resnik	the network-based measure of Resnik (1995) (3.4)
InvEdgeCount	the network-based measure "Inverted Edge Count" (3.2)
LeacockChodorow	the network-based measure of Leacock and Chodorow (1998)(3.3)
WuPalmer	the network-based measure of Wu and Palmer (1994) (3.7)
Lin	the network-based measure of Lin (1998a) (3.6)
JiangConrath	the network-based measure of Jiang and Conrath (1997) (3.5)
BDA	a corpus-based measure based on the Bag-of-words Distributional Analysis, i. e. the context window approach (Section 3.3.1)
BDA-k-cos	a corpus-based measure based on the context window of k words and the cosine (Section 3.3.1)
GlossVectors	a definition-based measure (Section 3.4)
ExtendedLesk	a definition-based measure (Section 3.4)
DefVectors	a definition-based measure based on the Vector Space Model (Section 2.3.1)
DefVectors-WktWiki	a definition-based measure based on the Vector Space Model and definitions of Wiktionary and Wikipeida (Section 3.4)
NP	noun phrase
LCS	lowest common subsumer (Section 3.2)
PMIIR	the web-based measure Pointwise Mutual Information Information Retrieval of Turney (2001) (Section 3.3.2)

PMIIR-Google	the web-based measure Pointwise Mutual Information Information Retrieval that relies on the Google search engine (Section 3.3.2)
NGD	the web-based measure Normalized Google Distance (Section 3.3.2)
NGD-Google	the web-based measure Normalized Google Distance that relies on the Google search engine (Section 3.3.2)
TF-IDF	Term Frequency - Inversed Document Frequency
Mean	a combination method: a mean of pairwise similarity scores (4.1)
MeanNnz	a combination method: a mean of similarity scores which have a non-zero value (4.2)
MeanNnz	a combination method: a mean of similarity scores transformed into Z-scores (4.3)
Median	a combination method: a median of pairwise similarities (4.4)
Max	a combination method: a maximum of pairwise similarities (4.5)
RankFusion	a combination method (4.6)
Logit	Logistic Regression (Section 4.2)
LogitL1	L1-regularized Logistic Regression (Section 4.2)
LogitL2	L2-regularized Logistic Regression (Section 4.2)
E15	a set of 15 similarity measures (Section 4.3)
Logit-E15	Logistic Regression trained on the set of 15 similarity measures *E15* (Sections 4.2, 4.3)
SVM	Support Vector Machine (Section 4.2)
SVM-linear-E15	Support Vector Machine with the linear kernel trained on the set of 15 similarity measures *E15* (Sections 4.2, 4.3)
CSA	Child Sexual Abuse, e. g. as in “CSA media” (Section 5.2)
TP	true positive
TN	true negative
FP	false positive

FN	false negative
Serelex	a system that performs search and visualization of semantically related words (Section 5.1)
NLP	Natural Language Processing
IR	Information Retieval
toplist	a list of the most semantically related words to a given query term

Introduction

La dernière chose qu'on trouve en laissant un ouvrage
est de savoir celle qu'il faut mettre la première.

– Blaise Pascal, *Pensée*, 1670

Semantic relations, such as synonyms, hypernyms and co-hyponyms proved to be useful for text processing applications, including text similarity, query expansion, question answering and word sense disambiguation. Such relations are practical because of the gap between lexical surface of the text and its meaning. Indeed, the same concept is often represented by different terms. However, existing resources often do not cover a vocabulary required by a given system. Manual resource construction is prohibitively expensive for many projects. On the other hand, precision of the existing extractors still do not meet quality of the hand-crafted resources. All these factors motivate the development of novel extraction methods.

This thesis deals with similarity measures for semantic relation extraction. The main research question we address, is how to improve precision and coverage of such measures. First, we perform a large-scale study the baseline techniques. Second, we propose four novel measures. One of them significantly outperforms the baselines, the others perform comparably to the state-of-the-art techniques. Finally, we successfully apply one of the novel measures in two text processing systems.

Chapter 1 begins with a description of the *object* of the research – semantic relations and resources. First, we define these objects formally and provide examples of resources commonly used in text processing systems, such as taxonomies, thesauri, lexical databases and ontologies. Second, we introduce the *subject* of the research – semantic relation extractors based on similarity measures. Finally, the chapter presents benchmarks designed to assess performance of this kind of extraction systems.

Chapter 2 deals with semantic similarity measures which rely on one resource (a corpus, a dictionary, etc.) and one extraction method (distributional analysis, lexico-syntactic patterns, etc.). The chapter begins with an overview of related work. Then, we describe three

experiments. In the first one, we propose a new similarity measure *SDA-MWE*, which stems from the syntactic distributional analysis, and apply it to the task of automatic thesaurus construction. The second one presents a new similarity measure *DefVectors* based on definitions. Finally, in the third experiment, we propose a new similarity measure *PatternSim*, which extracts "definitions" from a huge corpus with lexico-syntactic patterns. Three measures described in this chapter, perform comparably to the baselines, each with its pros and cons in terms of precision and coverage. We conclude that one way to significantly improve over the baselines could be to use the complementarity of different measures. This idea is developed in the next chapter.

Chapter 3 evaluates a wide range of baseline semantic similarity measures to identify their systematic advantages and disadvantages. The existing measures differ both in the kinds of information they use and in the ways this information is transformed into a similarity score. First, in this chapter, we present a comparative study of heterogeneous baseline measures. Several authors already compared existing approaches, but we perform a study on a large scale, as we compare 37 similarity measures based on semantic networks, text corpora, Web as a corpus, dictionaries and encyclopedia. Second, we go further than most of the surveys and compare the measures with respect to the semantic relation types they provide (hypernyms, meronyms, etc.). Our results suggest that the studied approaches are highly heterogeneous in terms of precision, coverage and semantic relation distributions. We address the problem of measure combination in the next chapter.

Chapter 4 describes several hybrid semantic similarity measures combining evidence from complementary sources. First, in this chapter, we present a systematic analysis of 16 baseline measures combined with 9 fusion methods and 3 measure selection techniques. Some attempts were already made to combine the baseline measures to improve the performance. However, we are first to propose hybrid similarity measures based on all main types of resources – semantic networks, text corpora, Web as a corpus, dictionaries and encyclopedia. Second, we describe several novel hybrid measure which combine 15 baselines with the supervised learning: *Logit-E15*, *C-SVM-linear-E15*, *C-SVM-radial-E15*, etc. They outperform all tested single and unsupervised hybrid measures by a large margin. Our results show that measures based on complementary sources of information indeed significantly outperform the baselines measures.

Chapter 5 presents two applications of semantic similarity measures to text processing. Both systems rely on the *PatternSim* measure introduced in Chapter 2. First, we describe *Serelex*, a system that given a query, provides a list of related terms and displays them in a form of an interactive graph or a set of images. Second, we describe a new short text categorization system, developed for processing filenames of P2P networks. We show that the relations

extracted with the similarity measure *PatternSim* improve the accuracy of the classification with the help of the *vocabulary projection* technique. Finally, in this chapter, we provide a list of further text processing applications, where semantic similarity measures may be useful. We conclude that the developed semantic similarity measures can indeed be practical for the real text processing systems.

Chapter 1

Semantic Relation Extraction: the Context and the Problem

This chapter introduces the context of the work, the motivation, the problem and the evaluation framework. Section 1.1 describes the *object* of the research – semantic relations and resources. We provide a formal model of semantic resource (Section 1.1.1) and illustrate it with several examples of manually-constructed structures, such as thesauri and ontologies (Section 1.1.2). Section 1.2 deals with the *subject* of the research – semantic relation extractors. First, we indicate limitations of the manually constructed resources in the context of Natural Language Processing (NLP) and Information Retrieval (IR) systems and motivate development of new extraction methods. Section 1.2.1 describes how relations could be extracted from text-based data (corpora, dictionaries, etc.). In this work, we are going to rely on a similarity-based relation extraction framework described in Section 1.2.2. The principal components of such extractors are specified. Finally, Section 1.2.3 presents the evaluation protocol used in this work.

1.1 Semantic Relations and Resources

There exist several types of semantic relations – synonyms, metonyms, antonyms, associations, etc. In the context of this work, we deal with synonyms, hypernyms and co-hyponyms (terms with a common hypernym, such as "Canon" and "Nikon"). We focus on them as they are useful for various NLP/IR applications, such as text similarity (Mihalcea et al., 2006; Tsatsaronis et al., 2010), word sense disambiguation (Patwardhan et al., 2003), query expansion (Hsu et al., 2006) and some others (see Chapter 5). Language processing systems need semantic relations because of the gap between the lexical surface of the text and its meaning. Two text documents may describe the same entity with different terms, such as

"computer", "PC", "machine" or "HP ProBook". If a system relies solely on the lexical representations, such as the bag-of-word model, it can provide sub-optimal results.

1.1.1 Definition

A set of typed semantic relations R between a set of terms C is a ternary relation $R \subset C \times T \times C$, where T is a set of relation types. A *typed semantic relation* $r \in R$ is a triple $\langle c_i, t, c_j \rangle$ linking two terms $c_i, c_j \in C$ with a semantic relation of type $t \in T$. In this work, we focus on synonyms, hypernyms and co-hyponyms: $T = \{$*syn, hyper, cohypo*$\}$. However, methods described in this thesis extract untyped relations. A set of untyped semantic relations R between a set of terms C is a binary relation $R \subset C \times C$. An *untyped semantic relation* $r \in R$ is a tuple $\langle c_i, c_j \rangle$ linking two terms with an unnamed semantic relation of type $t \in T = \{$*syn, hyper, cohypo*$\}$. Let us bring some examples of typed semantic relations:

- $\langle$*combustion gas, syn, exhaust gas*$\rangle$;
- $\langle$*discrete mathematics, hypo, science*$\rangle$;
- $\langle$*graph theory, cohypo, set theory*$\rangle$.

The following are examples of untyped semantic relations:

- $\langle$*car, vehicle*$\rangle$ – unnamed type syn;
- $\langle$*transport, subway*$\rangle$ – unnamed type $hyper$;
- $\langle$*physics, mathematics*$\rangle$ – unnamed type $cohypo$.

Different types of relations have different properties:

- reflexivity: $\forall c \in C : \quad \langle c, t, c \rangle$;
- symmetricity: $\forall c_i, c_j \in C : \quad \langle c_i, t, c_j \rangle \rightarrow \langle c_j, t, c_i \rangle$;
- transitivity: $\forall c_i, c_j, c_l \in C : \quad \langle c_i, t, c_j \rangle \wedge \langle c_j, t, c_l \rangle \rightarrow \langle c_i, t, c_l \rangle$.

Table 1.1 depicts the properties of the four types of relations we are dealing with in this thesis. Hyponyms and hypernyms are symmetric. For instance, "vehicle" is a hypernym of "bus" and "bus" is a hyponym of "vehicle". In this work, synonyms, co-hyponyms and hypernyms/hyponyms will be thus considered as symmetric relations.

Semantic Relation Type, t	**Reflexivity**	**Symmetricity**	**Transitivity**
synonymy, syn	yes	yes	no
hyponymy, $hypo$	no	no	yes
hypernymy, $hyper$	no	no	yes
co-hyponymy, $cohypo$	yes	yes	yes

Table 1.1: Properties of synonymy, hypernymy and co-hyponymy relations.

A *semantic resource* is a directed graph (C, R), which consists of:

- a set of nodes C, called *vocabulary*. Each node $c \in C$ represents a term such as "car", "vehicle" or "Ford Mustang".
- a set of edges R representing *semantic relations* between terms of the vocabulary C. An edge of a semantic resource $r \in R$ is a semantic relation, such as $\langle$*car, syn, vehicle*$\rangle$. If R is a set of typed relations ($R \subset C \times T \times C$), then edges of the graph are labeled with the relation types T (see Figure 1.1(a)). If R is a set of untyped relations ($R \subset C \times C$), then edges are unlabeled (see Figure 1.1(b)).

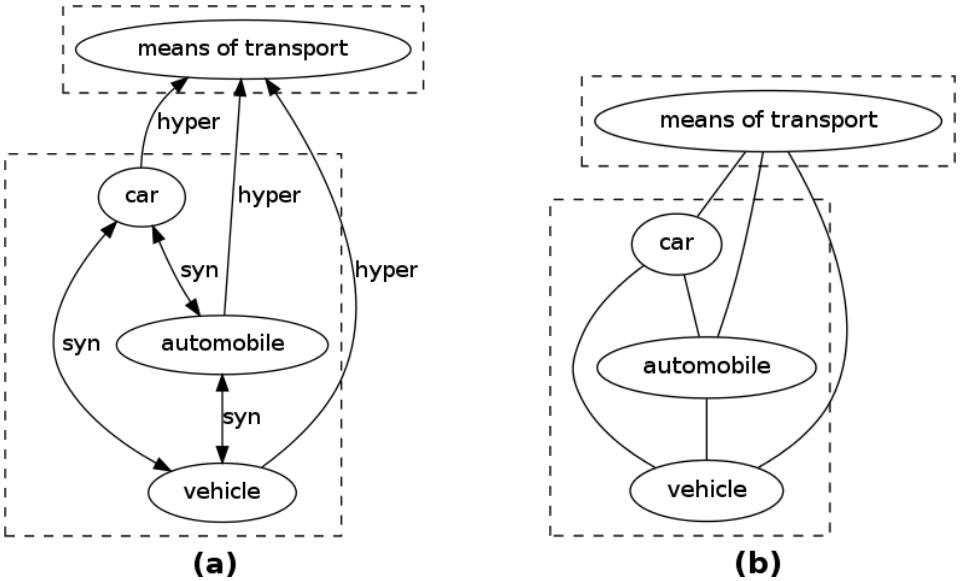


Figure 1.1: A semantic resource with typed (a) and untyped (b) semantic relations.

Figure 1.1 (a) depicts a semantic resource, which consists of four terms

$$C = \{\textit{means of transport, vehicle, car, automobile}\}, \tag{1.1}$$

and six typed semantic relations between them:

$$\begin{aligned} R = \{ \quad & \langle\textit{means of transport, hyper, car}\rangle, \langle\textit{means of transport, hyper, automobile}\rangle, \\ & \langle\textit{means of transport, hyper, vehicle}\rangle, \langle\textit{automobile, syn, car}\rangle, \\ & \langle\textit{car, syn, vehicle craft}\rangle, \langle\textit{vehicle, syn, automobile}\rangle\}. \end{aligned} \tag{1.2}$$

Figure 1.1 (b) depicts the same resource with unlabeled relations:

$$\begin{aligned} R = \{ \quad & \langle\textit{means of transport, car}\rangle, \langle\textit{means of transport, automobile}\rangle, \\ & \langle\textit{means of transport, vehicle}\rangle, \langle\textit{automobile, car}\rangle, \\ & \langle\textit{car, vehicle craft}\rangle, \langle\textit{vehicle, automobile}\rangle\}. \end{aligned} \tag{1.3}$$

Figure 1.2 (a) depicts a resource composed of 11 terms and 29 synonyms and hypernymy relations. Grouping synonyms into clusters, called *synsets* or *concepts*, let us represent the resource in a more compact way (see Figure 1.2 (b)). Note that the resource implicitly encodes many co-hyponyms:

- ⟨*air-cushion vehicle, cohypo, large vehicle*⟩, ⟨*truck, cohypo, electric vehicle*⟩,
- ⟨*bus, cohypo, electromobile*⟩, ⟨*van, cohypo, electromobile*⟩, etc.

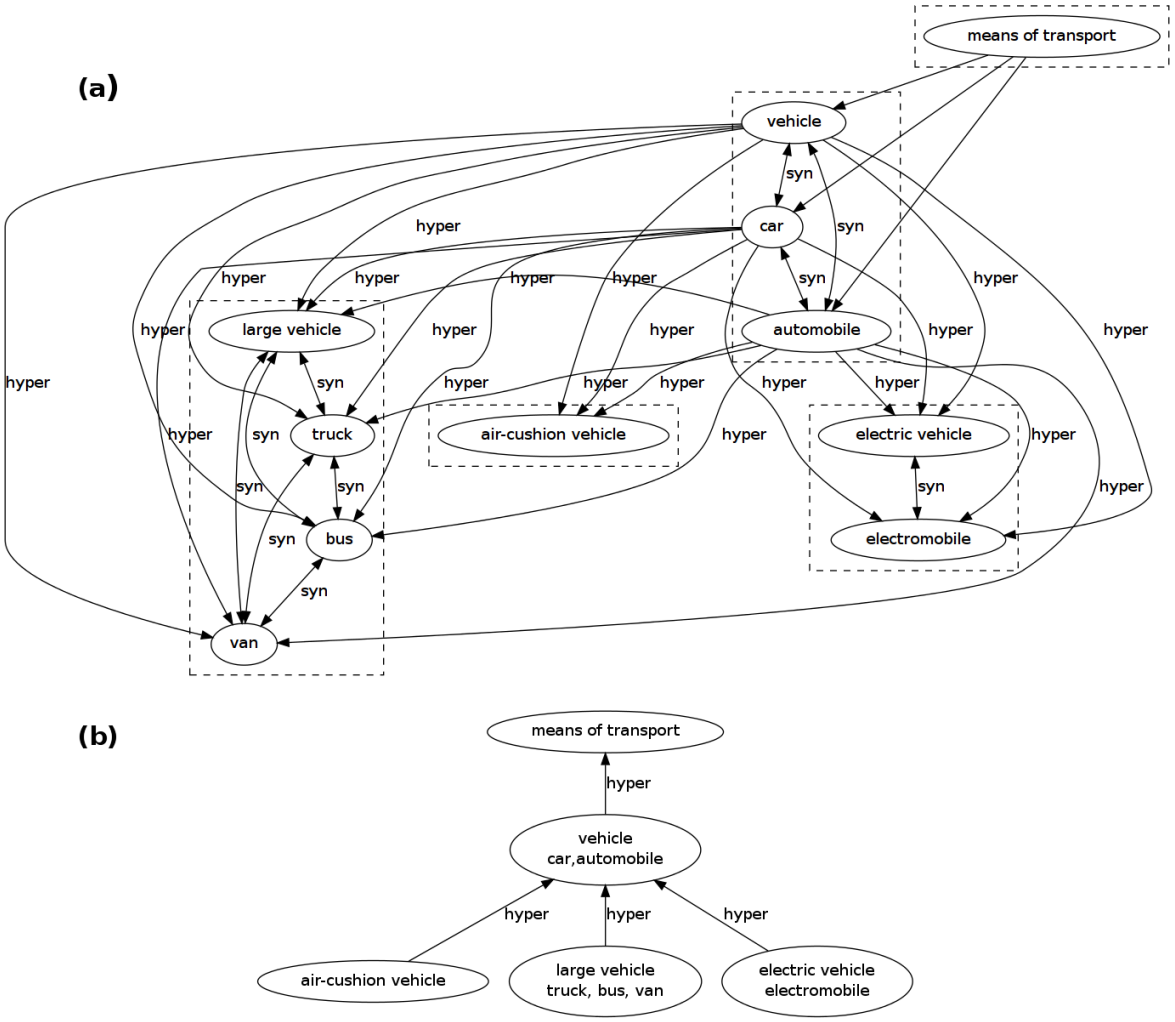


Figure 1.2: (a) a semantic resource with 29 relations; (b) the same resource with grouped synonyms.

A semantic resource (C, R) can be characterized with the following parameters:

- Language of its vocabulary C. In this work, we deal with English and French.
- Domain of its vocabulary C. We deal with the general and the political domains.
- Type of terms used in its vocabulary C. In this thesis, we deal with single words and multiword expressions.
- Size of its vocabulary $|C|$. In this work, we process vocabularies of different sizes: from 775 up to 419,751 terms.

- Number of semantic relations $|R|$. In this work, we deal with semantic resources with up to 11,251,240 semantic relations.
- Density of semantic relations, $\rho = \frac{|R|}{|C|}$.
- Structure of the resource (C, R), such as a tree, a network, etc. In this work, we set no restrictions on the structure of the resource.

1.1.2 Examples

Classification of conceptions with hierarchical and equivalence relations dates back to the Ancient Greeks. The first classification schemes were proposed by Aristotle (384-322 B.C.). "The Tree of Porphyry" is a tree structure of categories based on the Aristotle's work (Sowa, 1983). A biological taxonomy was proposed by Linnaeus in 1735. The first thesaurus was published by Roget in 1852. Peirce proposed a graphical notation called "existentional graphs" in 1909. Selz (1913) used graphs to represent conceptual hierarchies.

Modern semantic resources such as thesauri, ontologies or lexical databases differ in their structure, expressiveness and applications (see Figure 1.3). Below we overview several types of resources: controlled vocabularies, taxonomies, classification schemes, thesauri, subject headings, lexical databases and ontologies (see Table 1.2). The Taxonomy Warehouse catalog [1] lists around 690 taxonomies, thesauri, classification schemes and controlled vocabularies for 73 domains in 39 languages. The TONES repository [2] provides access to more than 200 ontologies. The Swoogle search engine [3] performs search over 2 millions Semantic Web documents such as OWL ontologies and SKOS thesauri.

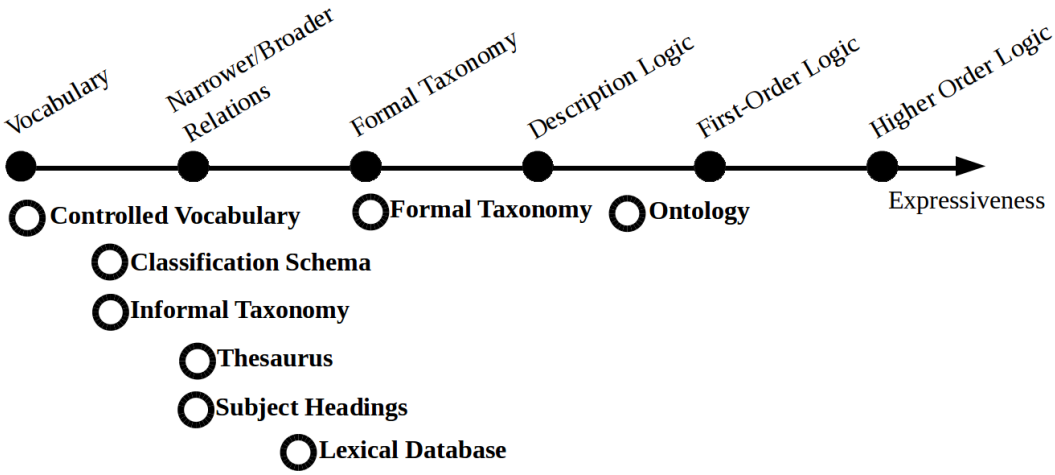


Figure 1.3: Expressiveness of various semantic resources.

[1] http://www.taxonomywarehouse.com/
[2] http://owl.cs.manchester.ac.uk/repository/
[3] http://swoogle.umbc.edu/

Semantic Resource	Semantic Relation Types
Controlled Vocabularies	synonyms
Taxonomies	hypernyms, co-hyponyms
Classification Schemes	hypernyms, co-hyponyms
Thesauri	synonyms, hypernyms, co-hyponyms, associations
Subject Headings	synonyms, hypernyms, co-hyponyms, associations
Lexical Databases	synonyms, hypernyms, co-hyponyms, associations, meronyms, as well as any other lexico-semantic relations
Ontologies	synonyms, hypernyms, co-hyponyms, as well as any other relations

Table 1.2: Comparison of the resources according to types of their semantic relations.

Controlled Vocabularies

A *controlled vocabulary* is the simplest semantic resource (C, R) composed of a list of terms C and their synonyms: $R \subset C \times T \times C$, where $T = \{syn\}$. In its simplest form, such resource may contain no synonyms at all: $R = \emptyset$. Tudhope et al. (2006) proposes the following definition:

> "Controlled vocabularies consist of terms, words from natural language selected as useful for retrieval purposes by the vocabulary designers. A term can be one or more words. A term is used to represent a concept. "

There exist several types of controlled vocabularies – authority files, glossaries, gazetteers, terminology dictionaries, synonym rings, dictionaries of synonyms and some others. Authority files such as Library of Congress Name Authority File [4] are lists of terms that are used to control variance in the names of countries, individuals and organizations. A glossary is a list of terms of a certain domain with their definitions. A gazetteer such as U.S. Census Gazeeter [5] is a list of geographical objects such as cities, rivers and mountains. Finally, a synonym ring is a structure where every concept has one preferred term and several alternative ones, often including misspellings and lexical variations (Hodge, 2000).

Taxonomies

A *taxonomy* is a semantic resource (C, R) composed of a lists of terms C organized into a hierarchy with a set of semantic relations: $R \subset C \times T \times C$, where $T = \{hyper\}$. Usually, a taxonomy organizes terms in a tree structure. There exist formal and informal taxonomies. Hierarchical relations of the formal taxonomies are transitive while relations of the informal taxonomies are not (Cimiano, 2006). Taxonomies has been used for a long time in Biology to categorize organisms, genus and species by biological type. Figure 1.4 depicts a part of a

[4] `http://authorities.loc.gov/`
[5] `http://www.census.gov/geo/www/gazetteer/gazette.html`

taxonomy of economic activities NACE [6]. Other examples of taxonomies include European Taxonomy of Skills Competences and Occupations [7], Cyc Taxonomies [8] and LexisNexis Taxonomies [9].

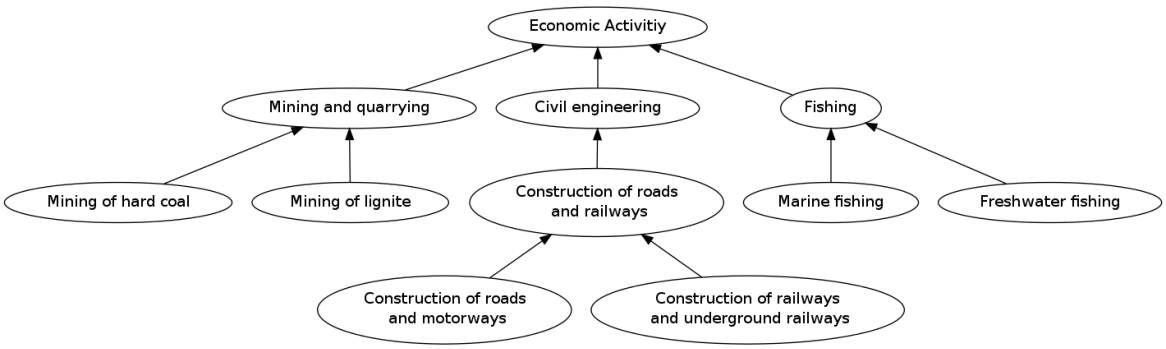


Figure 1.4: A part of the taxonomy of economical activities NACE.

Classification Schemes

Similarly to taxonomies, *classification schemes* are resources composed of terms C organized into a hierarchy with a set of relations $R \subset C \times T \times C$, where $T = \{hyper\}$. According to Hodge (2000), the goal of classification schemes is to organize documents according to the general topics. Examples of classification schemes include Dewey Decimal Classification [10], UNESCO nomenclature [11], Universal Decimal Classification [12], Harvard–Yenching Classification [13], and ACM Computing Classification System [14]. These classification schemes are widely used in traditional and digital libraries. Figure 3.3 depicts a part of the classification schema of the Library of Congress [15].

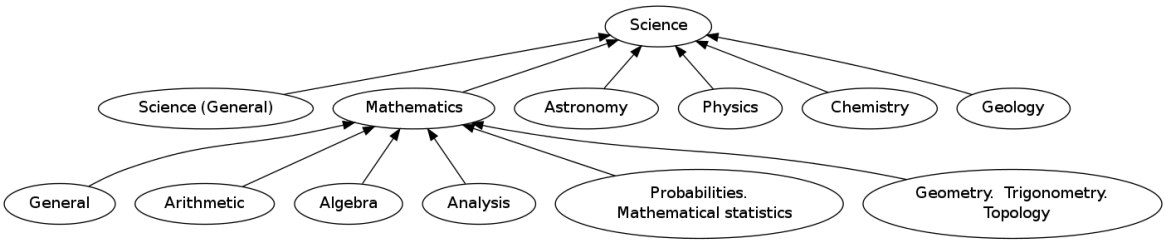


Figure 1.5: A part of the Library of Congress classification schema (LOC).

[6] http://ec.europa.eu/competition/mergers/cases/index/nace_all.htm
[7] http://ec.europa.eu/esco
[8] http://taxonomies.cyc.com/cyc/products
[9] http://www.lexisnexis.com/taxonomy/
[10] http://www.oclc.org/dewey/
[11] http://unesdoc.unesco.org/images/0008/000829/082946eb.pdf
[12] http://www.udcc.org/
[13] http://www.lib.unimelb.edu.au/collections/asian/Harvard-Yenching.html
[14] http://www.acm.org/about/class/1998/
[15] http://www.loc.gov/catdir/cpso/lcco/

Thesauri

A *thesaurus* is a resource (C, R) that contains terms C and hierarchical, equivalence and association relations between them: $R \subset C \times T \times C$. Hierarchical relations correspond to hypernyms, while equivalence relations roughly correspond to synonyms: $T = \{$*syn, hypo, assoc*$\}$. See Table 1.3 for details. Thesauri are used for information management and retrieval in restricted domains such as Medicine, Finance or Legislation. A thesaurus lists key terms of a domain and organizes them with relations. Figure 1.6 depicts a part of the Eurovoc thesaurus [16]. Other examples of thesauri include Agrovoc, International Classification of Diseases (ICD) [17], United Nations Thesaurus (UNBIS) [18], Unified Medical Language System (UMLS) [19], National Agriculture Library Agricultural Thesaurus (NAL) [20], Cedefop European Training Thesaurus [21], and Unesco Thesaurus [22].

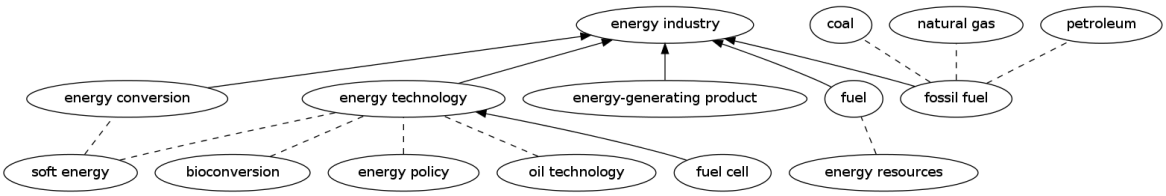


Figure 1.6: The Eurovoc thesaurus: the term "energy industry" and its semantic relations. Here, hypernyms are denoted with arrows and associations are denoted with dashed lines.

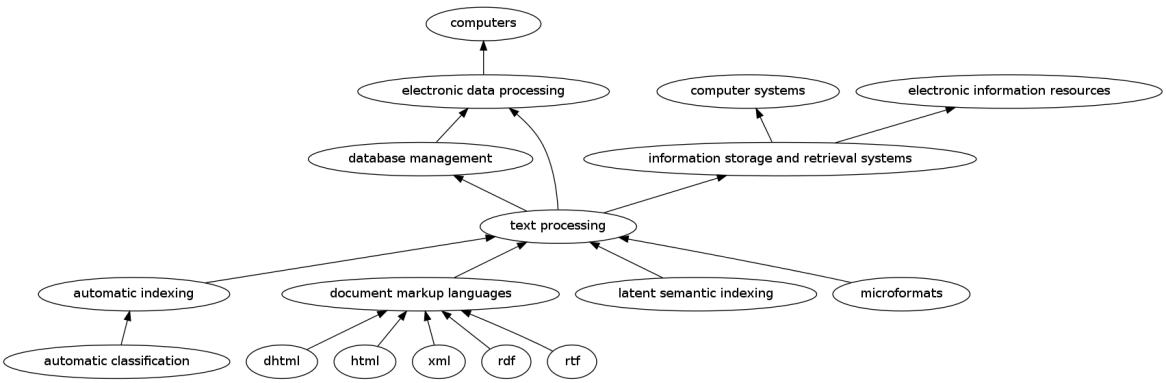


Figure 1.7: Library of Congress Subject Headings: term "text processing" and its semantic relations.

[16] http://eurovoc.europa.eu/
[17] http://www.who.int/classifications/icd/
[18] http://lib-thesaurus.un.org/
[19] http://www.nlm.nih.gov/research/umls/
[20] http://agclass.nal.usda.gov/
[21] http://libserver.cedefop.europa.eu/ett/
[22] http://databases.unesco.org/thesaurus/

Relation Type	Description	Example
equivalence	Synonymy	UN / United Nations
	Lexical variants	pediatrics / paediatrics
	Near synonymy	sea water / salt water
	References to Elements of Compound Terms	coal mining / coal / mining
hierarchical	Generic or IsA	birds / parrots
	Instance or IsA	sea / Mediterranean Sea
	Whole / Part	brain / brain stem
associative	Cause / Effect	accident / injury
	Process / Agent	velocity measurement / speedometer
	Process / Counter-agent	fire / flame retardant
	Action / Product	writing / publication
	Action / Property	communication / communication skills
	Action / Target	teaching / student
	Concept or Object / Property	steel alloy / corrosion resistance
	Concept or Object / Origins	water / well
	Concept or Object / Measurement Unit	chronometer / minute
	Raw material / Product	grapes / wine
	Discipline or Field / Object or Practitioner	neonatology / infant

Table 1.3: Semantic relation types between terms of a thesaurus specified in the international standard ANSI/NISO Z39.19-2005.

Subject Headings

Subject headings is a semantic resource (C, R) which organizes terms C with hierarchical, associative and equivalence relations: $R \subset C \times T \times C$, where $T = \{$*hypo, syn, assoc*$\}$. A term $c \in C$ is a *subject heading*. It can be a single word or a multi-word expression. Similarly to thesauri, subject headings are used for information management in a certain domain. Wellisch (1991) mentions that "most subject heading lists ... are characterized by the fact that they are much more loosely structured than thesauri and are, therefore, less effective for indexing or searching". However, some resources such as Medical Subject Headings (MeSH) [23] are used in the same way as thesauri. Other examples of subject headings include Library of Congress Subject Headings (LCSH) [24], Canadian Subject Headings (CSH) [25], RAMEU subject headings [26], and Schools Catalogue Information Service subject headings (SCIS) [27]. Figure 1.7 depicts a subject heading *text processing* from the LCSH and its semantic relations.

Lexical Databases

A *lexical database* is a triple $(C, \mathcal{S}, \mathcal{R})$, where C is a vocabulary, $\mathcal{S}$ is a set of synsets, $\mathcal{R}$ is a set of semantic relations between synsets $\mathcal{S} \times T \times \mathcal{S}$ and T is set of semantic relation

[23] `http://www.nlm.nih.gov/mesh/`
[24] `http://www.loc.gov/aba/cataloging/subject/`
[25] `http://www.collectionscanada.gc.ca/csh/`
[26] `http://www.cs.vu.nl/STITCH/rameau/`
[27] `http://www2.curriculum.edu.au/scis/subject_headings.html`

types. Vocabulary of a lexical database often contain ambiguous terms, unlike thesauri, taxonomies and other resources described above. A synset $s \in \mathcal{S}$ is a set of mutual synonyms: $s = \{c_i, \ldots, c_j\} : \forall c_i, c_j \in s \Rightarrow \langle c_i, syn, c_j \rangle$. For instance, the synset *engineer* may be composed of three terms: $s = \{$*engineer, applied scientist, technologist*$\}$. Each synset can have a definition such as *"a person who uses scientific knowledge to solve practical problems"*. Different senses of an ambiguous term, such as "python" or "jaguar", are represented by different synsets.

Figure 1.8 depicts a part of the WordNet (Miller, 1995b) lexical database. Lexical databases often contain a big number of relation types T. For instance, WordNet links "noun synsets" with the following relations: hypernymy, coordination, holonymy, meronymy and instance-of; "verb synsets" are linked with the following relations: hypernymy, troponymy, entailment and coordination; "adjective synsets" are linked with the following relations: related nouns, similar to, participle of verb and antonymy.

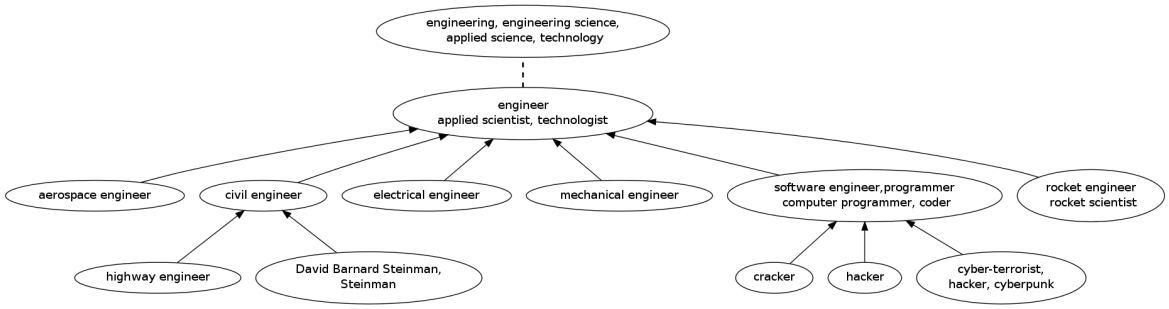


Figure 1.8: Lexical database WordNet: synset engineer and its semantic relations.

A lexical database $(C, \mathcal{S}, \mathcal{R})$ can be considered as a semantic resource (C, R) composed of a set of terms C and a set of semantic relations between them $R = R_{syn} \cup R_T$. Here $R_{syn} \subset C \times T \times C, T = syn$ is a set of synonyms generated from the synsets $\mathcal{S}$:

$$R_{syn} = \{\langle c_i, syn, c_j \rangle : \exists k, c_i \in s_k, c_j \in s_k, \}, c_i \in C, c_j \in C, s_k \in \mathcal{S}, \tag{1.4}$$

and $R_T \subset C \times T \times C$ is a set of semantic relations between terms generated from relations between the synsets $\mathcal{R} \subset \mathcal{S} \times T \times \mathcal{S}$ (see Figure 1.9):

$$R_T = \{\langle c_i, t, c_j \rangle : \exists \langle s_i, t, s_j \rangle, c_i \in s_i, c_j \in s_j\}, c_i \in C, c_j \in C, s_i \in \mathcal{S}, s_j \in \mathcal{S}. \tag{1.5}$$

According to Tudhope et al. (2006), WordNet is the most widespread lexical database. It contains 155,287 terms organized in 117,659 synsets [28]. Stamou et al. (2002) points out than more that 50 WordNet-like lexical databases are under construction for more than 40

[28] `http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html`

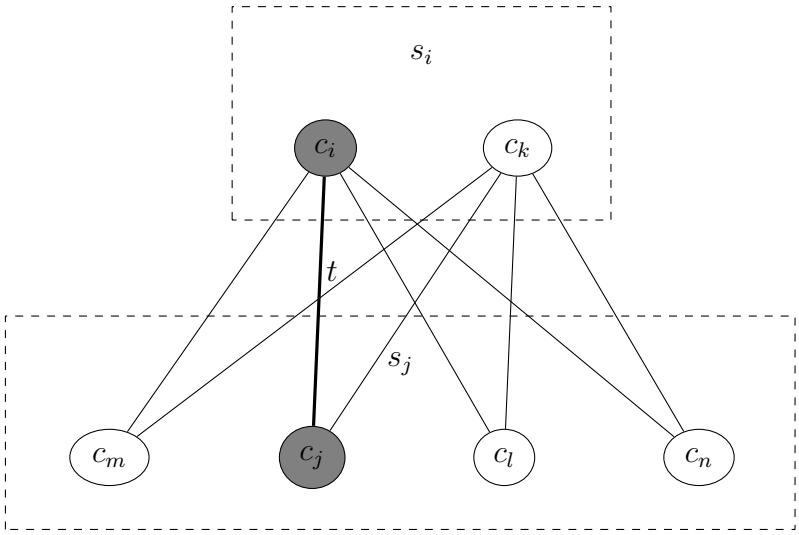


Figure 1.9: Decomposition of a relation between two synsets $\langle s_i, t, s_j \rangle \in \mathcal{R}$ into several semantic relations between terms $\{\langle c_i, t, c_j \rangle\} \in R$.

languages. Examples include EuroWordNet (Ellman, 2003), GermaNet (Hamp and Feldweg, 1997), WOrdnet Libre du Français (WOLF) (Sagot and Fišer, 2008), BalkaNet (Stamou et al., 2002) and Russian WordNet (Balkova et al., 2004). Furthermore, there exist several multilingual lexical databases such as MultiWordNet (Bentivogli et al., 2002), UWN (de Melo and Weikum, 2009) and BabelNet (Navigli and Ponzetto, 2010).

Ontologies

An ontology is a general knowledge representation model. According to Cimiano (2006), an *ontology* is a structure $\langle \mathcal{C}, \leq_{\mathcal{C}}, \mathcal{R}, \sigma_{\mathcal{R}}, \leq_{\mathcal{R}}, \mathcal{A}, \sigma_{\mathcal{A}}, \mathcal{T} \rangle$ consisting of a set of classes $\mathcal{C}$, relations $\mathcal{R}$, attributes $\mathcal{A}$ and datatypes $\mathcal{T}$. The semi-upper lattice $\leq_{\mathcal{C}}$ defines class type hierarchy and the partial order $\leq_{\mathcal{R}}$ defines relation hierarchy. The functions $\sigma_{\mathcal{R}}$ and $\sigma_{\mathcal{A}}$ are used to get correspondingly relations and attributes by their identifier. In addition, an ontology has an axiom system $\mathcal{S}$ and a lexicon $\langle \mathcal{S}_{\mathcal{C}}, \mathcal{S}_{\mathcal{R}}, \mathcal{S}_{\mathcal{A}}, Ref_{\mathcal{C}}, Ref_{\mathcal{R}}, Ref_{\mathcal{A}} \rangle$, where $\mathcal{S}_{\mathcal{C}}$, $\mathcal{S}_{\mathcal{R}}$, $\mathcal{S}_{\mathcal{A}}$ are lexical labels of classes, relations and attributes, correspondingly. The relations $Ref_{\mathcal{C}}, Ref_{\mathcal{R}},$ and $Ref_{\mathcal{A}}$ are lexical references for classes, relations and attributes, correspondingly. These references define lexical labels for classes, relations and attributes. A knowledge base of an ontology defines set of instances $\mathcal{I}$ of these classes and a set of lexical labels $\mathcal{S}_{\mathcal{I}}$ for the instances $\mathcal{I}$. An ontology with a lexicon represents *ontological* knowledge about a domain, while a knowledge base represents *factual* knowledge about the domain. Thus, ontology can represent not only lexico-semantic knowledge. However, semantic relations form the basis of each ontology. The concepts and their labels ($C, \mathcal{S}_{\mathcal{C}}$ and Ref_C) define synonyms, while the concept hierarchy $\leq_{\mathcal{C}}$ defines hypernyms.

Upper ontologies such as OpenCyc [29], DOLCE [30], GFO [31], SUMO [32], BFO [33] or YAMATO [34] represent common knowledge. Figure 1.10 illustrates a part of the SUMO upper ontology. Domain ontologies such as Disease Ontology [35], e-Business Model Ontology, Geopolitical Ontology [36], Plant Ontology [37], Gene Ontology [38], Customer Complaint Ontology [39], and Ontology for Biomedical Investigations [40] represent domain-specific knowledge. Cross-domain ontologies such DBPedia Ontology [41] contain a mix of high-level and domain-specific concepts. Lexical ontologies such as OntoWordNet (Gangemi et al., 2003) or the OWL representation of WordNet (Van Assem et al., 2006) represent lexico-semantic knowledge in the ontology format.

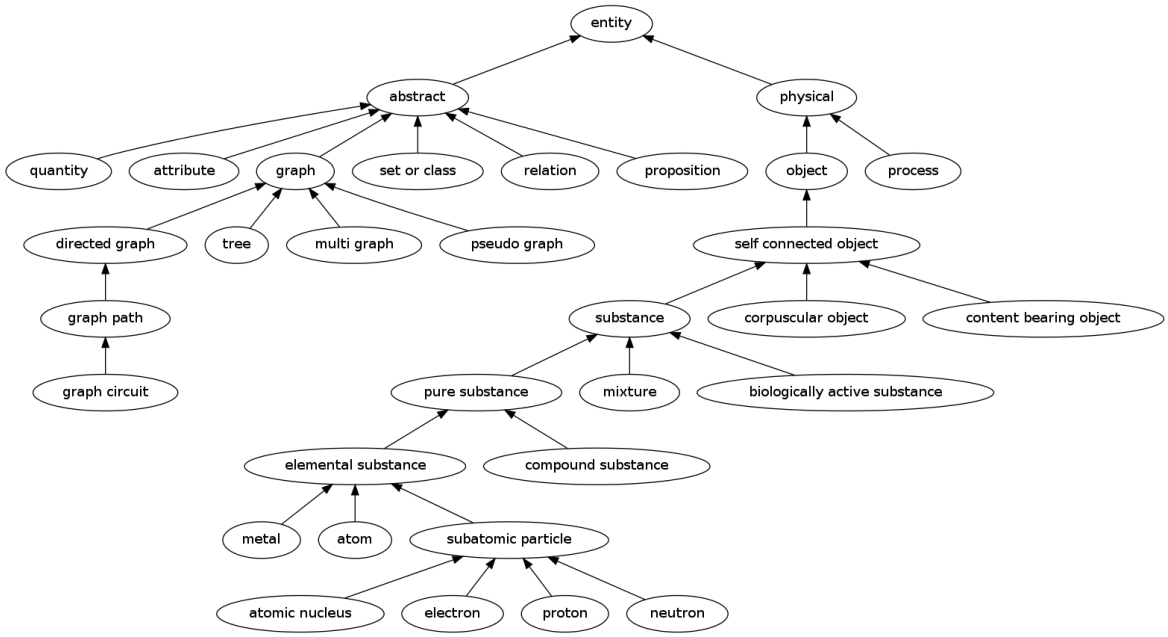


Figure 1.10: SUMO upper ontology: a part of the class hierarchy.

[29] http://www.opencyc.org/doc/
[30] http://www.loa-cnr.it/DOLCE.html
[31] http://www.onto-med.de/ontologies/gfo/
[32] http://www.ontologyportal.org/
[33] http://www.ifomis.org/bfo/
[34] http://www.ei.sanken.osaka-u.ac.jp/hozo/onto_library/upperOnto.htm
[35] http://do-wiki.nubic.northwestern.edu/
[36] http://aims.fao.org/geopolitical.owl
[37] http://www.plantontology.org/
[38] http://www.geneontology.org/
[39] http://www.jarrar.info/CContology/
[40] http://obi-ontology.org/
[41] http://dbpedia.org/Ontology

1.2 Semantic Relation Extraction

Semantic relations are useful for various text processing applications. However, existing resources are often not available for a given application, domain or language. One solution is to craft a required resource manually, for instance as described in the international standards ISO 2788, ISO 5964 and ANSI/NISO Z39.19-2005. However, manual construction is very expensive and time-consuming. Establishing semantic relations between terms is a subjective error-prone process, which involves a big amount of human labor. Furthermore, updating resource is also a manual time-consuming process. All these factors limit use of semantic resources in the NLP systems and/or hamper the performance of these systems. One solution to these problems is to extract semantic relations from texts. However, the quality of the automatically extracted relations is still lower than the quality of manually constructed relations (see Grefenstette (1994), Curran and Moens (2002), Heylen et al. (2008), Section 2.1 and Section 3.1). This motivates the development of new relation extraction techniques described in this thesis.

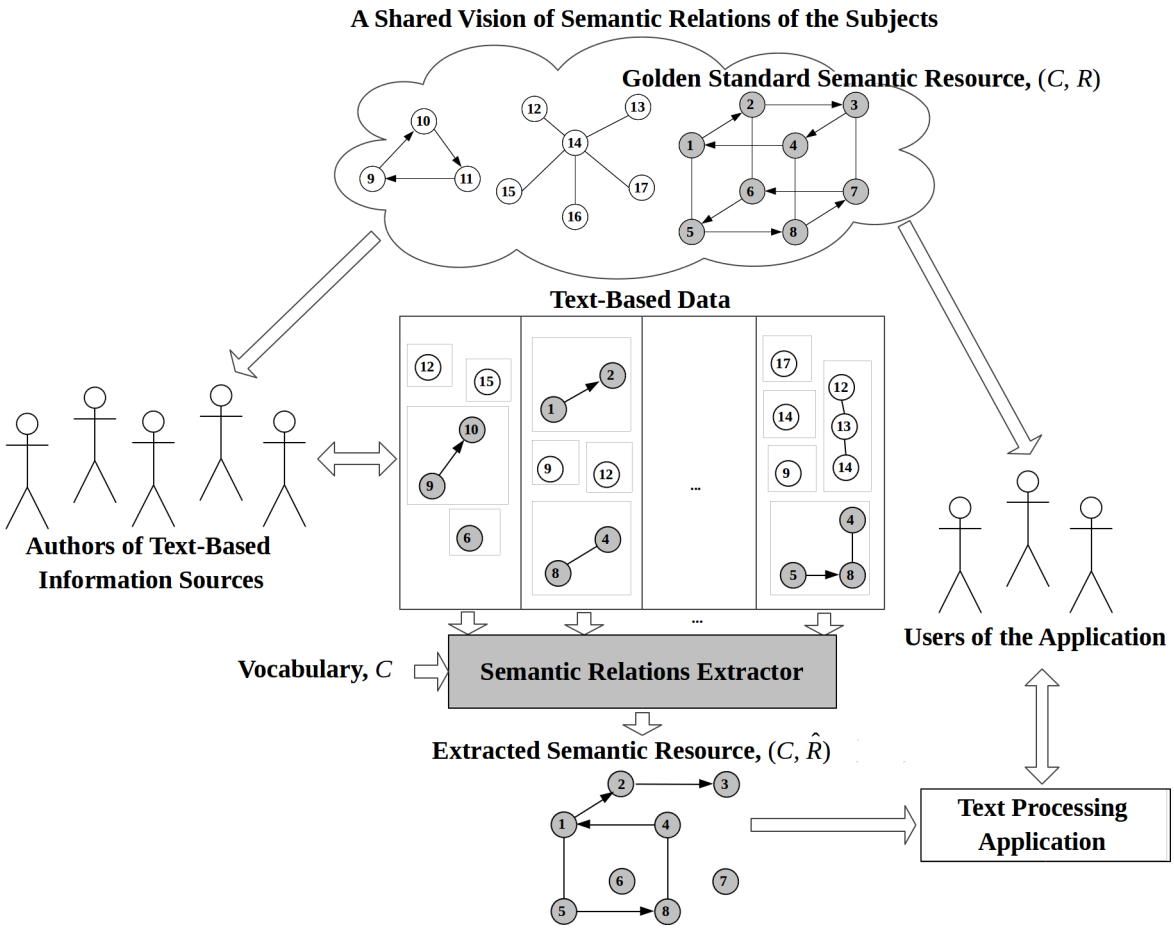


Figure 1.11: Extraction of semantic relations and using them in a text processing application.

1.2.1 Extraction Process

Let (C, R) be a manually constructed semantic resource, where $R \subseteq C \times C$ is a set of synonyms, hypernyms, hyponyms and co-hyponyms of terms C. A relation extractor aims to construct a set of relations $\hat{R} \subseteq C \times C$ as close to the golden standard resource R as possible in terms of precision and recall:

$$\hat{R}^* = \arg\max_{\hat{R}} \frac{Precision(R, \hat{R}) \cdot Recall(R, \hat{R})}{Precision(R, \hat{R}) + Recall(R, \hat{R})}, \text{where} \tag{1.6}$$

$$Precision(R, \hat{R}) = \frac{|R \cap \hat{R}|}{|\hat{R}|}, Recall(R, \hat{R}) = \frac{|R \cap \hat{R}|}{|R|}. \tag{1.7}$$

According to Philippovich and Prokhorov (2002), the conditions of automatic knowledge acquisition are the following:

- existence of a subject possessing the required knowledge;
- existence of a knowledge representation format;
- a method which extracts the knowledge of the subject to the specified format.

Information about semantic relatedness is implicitly encoded in texts. Therefore, the analysis of a big amount of texts should reveal these relationships (see Figure 1.11). We rely on several kinds of text-based data: text corpora, dictionaries, encyclopedia, Web corpus and semantic networks. Indeed, most extraction methods rely on text (Grefenstette, 1994; Landauer and Dumais, 1997; Caraballo, 1999; Curran and Moens, 2002; Cimiano, 2006; Moens, 2006; Auger and Barrière, 2008). However, some information about semantic relatedness can be mined from other sources such as query logs (Baeza-Yates and Tiberi, 2007), folksonomy structure (Strube and Ponzetto, 2006; Hadj Taieb et al., 2012) or hyperlink structure (Nakayama et al., 2007). In this work, we use semantic resource (C, R) as a knowledge representation format (see Section 1.1.1). Finally, the method which encodes knowledge of the subject to this format is described in Section 1.2.2.

Once semantic relations are extracted, they are used in a text processing application (see Figure 1.11). The extracted resource can be useful for the applications because of two reasons. First, texts processed by the system are similar to the texts used for the extraction. Thus, the resource can be used to generate plausible variations of the texts. Second, authors of the texts used for extraction and users of the application share the same notion of semantic relatedness. Thus, if the system uses the resource, the result should be plausible for the users. The extracted resource is often incomplete because of two reasons. First, the extractor always deals with a text which does not cover some terms. Second, some basic facts are rarely or never expressed in text.

1.2.2 Similarity-Based Extraction

In this work, we use an extraction method based on a *semantic similarity measure* and a nearest neighbors procedure (see Figure 1.12). The extractor takes as an input a vocabulary C and some text-based data. It outputs a set of semantic relations between input terms: $\hat{R} \subseteq C \times C$. First, a *feature extractor* represents each input term $c_i \in C$ as a numerical vector $\mathbf{f}_i$. Next, a *similarity measure* calculates a $|C| \times |C|$ term-term similarity matrix $\mathbf{S}$ from the feature matrix $\mathbf{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_n)^T$. The similarity scores are mapped to the interval $[0; 1]$ by a *normalizer* as follows: $\acute{\mathbf{S}} = \frac{\mathbf{S} - \min(\mathbf{S})}{\max(\mathbf{S}) - \min(\mathbf{S})}$. The normalizer also transforms dissimilarity scores into similarity scores, if needed (see examples below). Finally, a k-NN procedure calculates untyped semantic relations $\hat{R}$ between terms C from the similarity scores $\acute{\mathbf{S}}$, using some thresholding strategy. The extractor recalls pairs of untyped semantic relations (see Section 1.1.1): $\hat{R} \subseteq C \times C$. Nonetheless, we suppose that the system must retrieve a mix of synonyms, hypernyms and co-hyponyms and evaluate it accordingly (see Section 1.2.3).

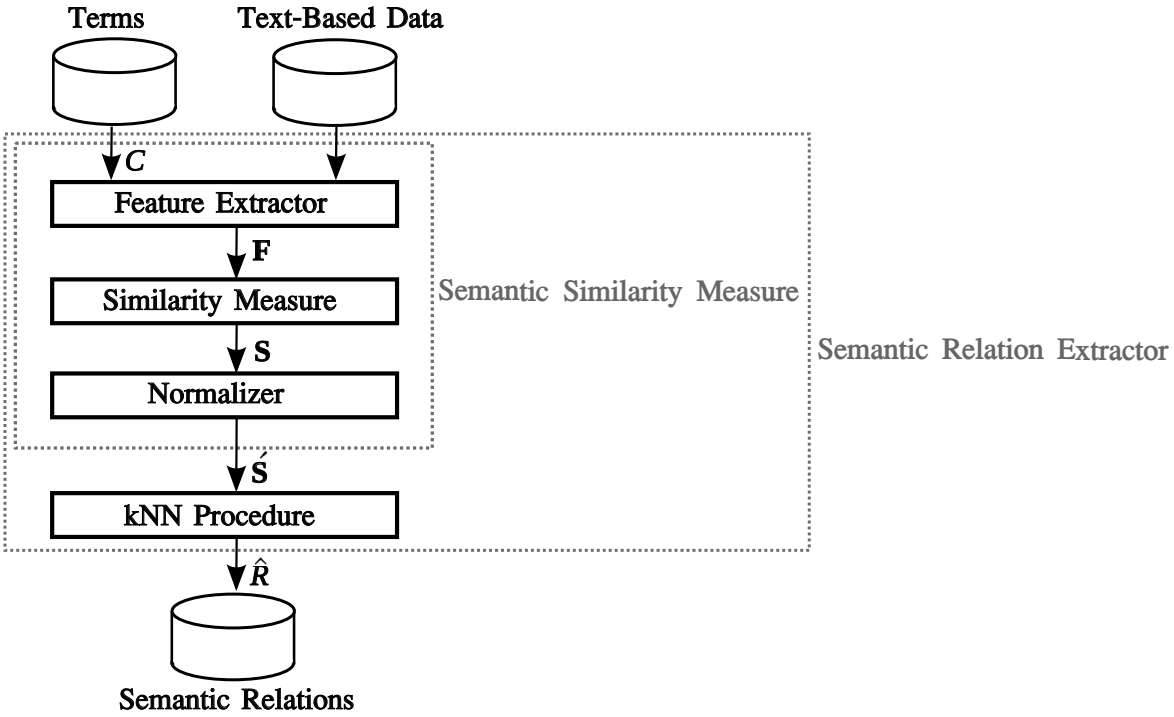


Figure 1.12: Structure of a similarity-based semantic relation extractor.

Thus, a *similarity measure* is a function which calculates a similarity score from a provided numerical vector. On the other hand, a *semantic similarity measure* is a method (or a system), which relies on features and a similarity measure adapted to the relation extraction task. In this work, we do not seek for general similarity measures. We rather look for semantic similarity measures practical for extraction of synonyms, hyponyms and co-hyponyms. The following sections provide details on the similarity measures, semantic similarity measures and nearest neighbor procedures.

Similarity and Dissimilarity Measures

According to Tan (2006), a *similarity measure* "is a numerical measure of the degree the two objects are alike", while a *dissimilarity measure* "is a numerical measure of the degree to which the two objects are different". Both similarity and dissimilarity scores are scalars in range $[0; 1]$ or $[0; \infty]$. Similarity and dissimilarity measures are symmetrically related. Two similar objects i and j will have a high similarity score s_{ij} and a low dissimilarity score d_{ij}. A similarity score can be transformed into dissimilarity and vice versa as follows:

- if $d_{ij} \in [0; 1]$, then $s_{ij} = 1 - d_{ij}$, where $s_{ij} \in [0; 1]$;
- if $s_{ij} \in [0; 1]$, then $d_{ij} = 1 - s_{ij}$, where $d_{ij} \in [0; 1]$;
- if $d_{ij} \in [0; \infty]$, then $s_{ij} = 1 - \frac{d_{ij} - \min_{i,j}(d_{ij})}{\max_{i,j}(d_{ij}) - \min_{i,j}(d_{ij})}$, where $s_{ij} \in [0; 1]$;
- if $s_{ij} \in [0; \infty]$, then $d_{ij} = 1 - \frac{s_{ij} - \min_{i,j}(s_{ij})}{\max_{i,j}(s_{ij}) - \min_{i,j}(s_{ij})}$, where $d_{ij} \in [0; 1]$.

Distances (or *metrics*) are dissimilarity measures which have four following properties (Tan, 2006; Theodoridis and Koutroumbas, 2009; Poole, 2010):

1. positivity: $\forall i, j : 0 \leq d_{ij} \leq 1$;
2. symmetry: $\forall i, j : d_{ij} = d_{ji}$;
3. identity: $d_{ij} = 0$ iff $i = j$;
4. triangle inequality: $\forall i, j : d_{ik} \leq d_{ij} + d_{jk}$.

Minkowski Distance takes as an input two n-dimensional vectors $\mathbf{f}_i$ and $\mathbf{f}_j \in \mathbb{R}^n$:

$$d_{ij} = \left(\sum_{k=1}^{n} |f_{ik} - f_{jk}|^l \right)^{\frac{1}{l}}, \text{ where } \mathbf{f}_i = (f_{i1}, \ldots, f_{in})^T. \tag{1.8}$$

The metric with $l = 1$ is known as L_1 *norm* (or *Manhattan Distance*):

$$d_{ij} = \left(\sum_{k=1}^{n} |f_{ik} - f_{jk}|^1 \right)^{\frac{1}{1}} = \sum_{k=1}^{n} |f_{ik} - f_{jk}| = ||\mathbf{f}_i - \mathbf{f}_j||_1. \tag{1.9}$$

The metric with $l = 2$ is known as L_2 *norm* (or *Euclidian Distance*):

$$d_{ij} = \left(\sum_{k=1}^{n} |f_{ik} - f_{jk}|^2 \right)^{\frac{1}{2}} = \sqrt{\sum_{k=1}^{n} (f_{ik} - f_{jk})^2} = ||\mathbf{f}_i - \mathbf{f}_j||_2 = ||\mathbf{f}_i - \mathbf{f}_j||. \tag{1.10}$$

The metric with $l = \infty$ is known as *L∞ norm* (or *Maximum Distance*):

$$d_{ij} = \lim_{l \to \infty} \left(\sum_{k=1}^{n} |f_{ik} - f_{jk}|^l \right)^{\frac{1}{l}} = \max_{k=1,n} |f_{ik} - f_{j1}| = ||\mathbf{f}_i - \mathbf{f}_j||_\infty. \tag{1.11}$$

A unit circle is a set of all unit vectors for a given norm. Different norms impose different unit circles (see Figure 1.13 (a)). Each of the distances mentioned above rely on the norm of vector difference: $d_{ij} = ||\mathbf{f}_i - \mathbf{f}_j||_l$. Figure 1.13 (b) illustrates dissimilarity scores calculated with three described above norms. For higher l, absolute values are lower: $\forall i, j : d_{ij}^{L1} \geq d_{ij}^{L2} \geq d_{ij}^{L\infty}$.

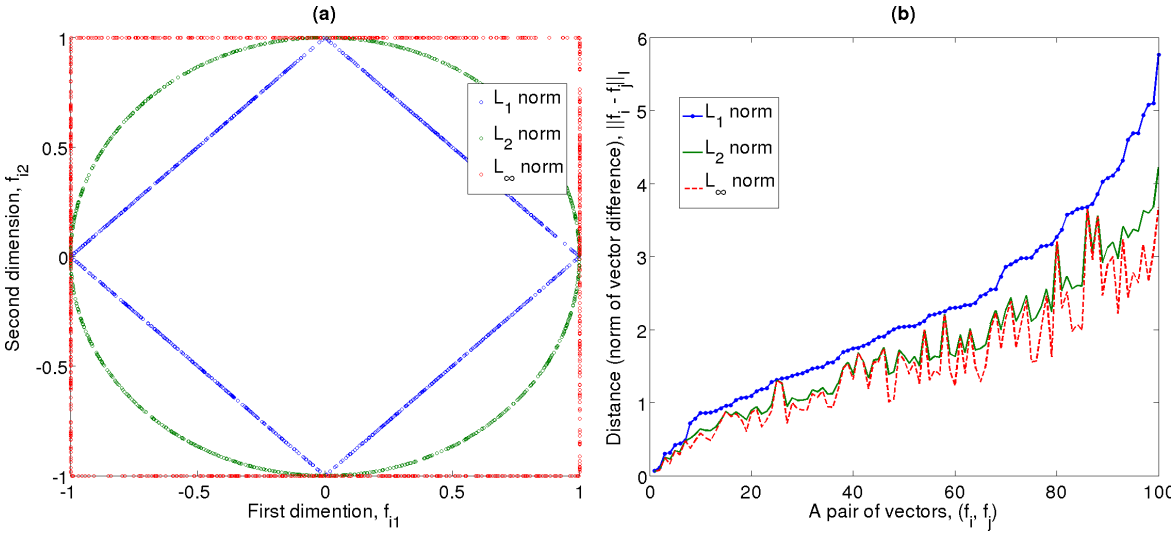


Figure 1.13: (a) unit circles of L_1, L_2 and L_∞ norms: $||\mathbf{f}_i||_l = 1$; (b) distances between random vectors calculated with L_1, L_2 and L_∞ norms.

Another well-known metric is *Mahalanobis distance* (Mahalanobis, 1936):

$$d_{ij} = (\mathbf{f}_i - \mathbf{f}_j)\boldsymbol{\Sigma}^{-1}(\mathbf{f}_i - \mathbf{f}_j)^T. \tag{1.12}$$

Here $\boldsymbol{\Sigma}$ is the covariance matrix of the $n \times m$ data matrix $(\mathbf{f}_1, \ldots, \mathbf{f}_m)^T$, where covariance $\sigma_{ij} \in \boldsymbol{\Sigma}$ is defined as follows:

$$\sigma_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} (f_{ik} - \bar{f}^i)(f_{jk} - \bar{f}^j), \text{ where } \bar{f}^k = \frac{1}{m} \sum_{i=1}^{m} f_{ik}. \tag{1.13}$$

Another common dissimilarity measure is *Jenson-Shannon Divergence* between two distributions (Lee, 1999; Jurafsky and Martin, 2009):

$$d_{ij} = D(P(i)||\frac{P(i) + P(j)}{2}) + D(P(j)||\frac{P(i) + P(j)}{2}), \tag{1.14}$$

where $D(P(i)||P(j))$ is *Kulback-Leibler Divergence* (or relative entropy) between probability distributions $P(i)$ and $P(j)$ (Kullback and Leibler, 1951):

$$D(P(i)||P(j)) = \sum_{k=1} P(i,k) \log \frac{P(i,k)}{P(j,k)}. \tag{1.15}$$

Jenson-Shannon Divergence quantify dissimilarity of objects i and j as the dissimilarity of their probability distributions $P(i)$ and $P(j)$. Therefore, to use this measure, one should transform the numerical vectors $\mathbf{f}_i$ and $\mathbf{f}_j$ to a probability distributions $P(i)$ and $P(j)$, so $\forall i : \sum_k f_{ik} = \sum_k P(i,k) = 1$.

In this work, we mostly deal with similarity measures. In contrast to the distances, they do not have the triangle inequality property:

1. positivity: $\forall i, j : 0 \leq s_{ij} \leq 1$;
2. symmetry: $\forall i, j : s_{ij} = s_{ji}$;
3. identity: $s_{ij} = 1$ iff $i = j$.

Examples of similarity measures between two numerical vectors include *Cosine Similarity* (Tan, 2006; Corral, 2008; Jurafsky and Martin, 2009):

$$s_{ij} = \cos(\mathbf{f}_i, \mathbf{f}_j) = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{||\mathbf{f}_i||||\mathbf{f}_j||}, \text{ where } \mathbf{f}_i \cdot \mathbf{f}_j = \sum_{k=1}^{m} f_{ik} f_{jk}; \tag{1.16}$$

and *Tanimoto Coefficient* (Rogers and Tanimoto, 1960; Theodoridis and Koutroumbas, 2009):

$$s_{ij} = \frac{\mathbf{f}_i \cdot \mathbf{f}}{||\mathbf{f}_i||^2 + ||\mathbf{f}_j||^2 - \mathbf{f}_i \cdot \mathbf{f}_j}. \tag{1.17}$$

Tanimoto Coefficient for the real value vectors (also called *Extended Jaccard Coefficient*) stems from the *Jaccard Index*, a similarity measure for binary vectors (Jaccard, 1901):

$$s_{ij} = \frac{m_{11}}{m_{01} + m_{10} - m_{11}} = \frac{\sum_{k=1}^{n}(f_{ik} \wedge f_{jk})}{\sum_{k=1}^{n}(f_{ik} \vee f_{jk})}, \text{ where } \mathbf{f}_i, \mathbf{f}_j \in \{0,1\}^n. \tag{1.18}$$

Here m_{00} is the number of 0-0 matches of $\mathbf{f}_i$ and $\mathbf{f}_j$, m_{01} is the number of 0-1 matches of $\mathbf{f}_i$ and $\mathbf{f}_j$, etc.

Dice Coefficient (or *Sorenson Similarity Index*) is a similarity measure for binary vectors closely related to the Jaccard Index (Dice, 1945; Sorenson, 1948):

$$s_{ij}^{dice} = \frac{2 \cdot s_{ij}^{jaccard}}{1 + s_{ij}^{jaccard}} = \frac{2 \cdot \sum_{k=1}^{n}(f_{ik} \wedge f_{jk})}{\sum_{k=1}^{n} f_{ik} + \sum_{k=1}^{n} f_{jk}}, \text{ where } \mathbf{f}_i, \mathbf{f}_j \in \{0,1\}^n. \tag{1.19}$$

Curran (2003) proposed the following version of the *Dice Coeffieient* for the real valued vectors:

$$s_{ij} = \frac{2 \cdot \sum_{k=1}^{n} \min(f_{ik}, f_{jk})}{\sum_{k=1}^{n} (f_{ij} + f_{jk})}. \tag{1.20}$$

Pearson Correlation can also be used as a similarity measure as following:

$$s_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}} = \frac{\sum_{k=1}^{n}(f_{ik} - \bar{f}^{i})(f_{jk} - \bar{f}^{j})}{\sum_{k=1}^{n}(f_{ik} - \bar{f}^{i})^2 \sum_{k=1}^{n}(f_{jk} - \bar{f}^{j})^2}. \tag{1.21}$$

However, correlation values are in the range $[-1; +1]$ and thus should be transformed into the interval $[0; 1]$, e. g. as following:

$$s'_{ij} = \frac{s_{ij} - \min_{i,j}(s_{ij})}{\max_{i,j}(s_{ij}) - \min_{i,j}(s_{ij})}, \text{ where } s_{ij} \in [-1; +1] \text{ and } s'_{ij} \in [0; 1]. \tag{1.22}$$

In this thesis, we experiment with different similarity measures. We rely on *Cosine Similarity* (Sections 2.2.2, 2.3.1, 3.3.1, 3.3.3, 3.4), *Jaccard Index* (Section 2.3), *Extended Jaccard Coefficient* (Section 3.3.1), *Euclidean Distance* (Section 3.3.1), *Manhattan Distance* (Section 3.3.1), as well as some other specific unsupervised (Sections 2.3.1, 2.4.2, 3.2, 3.3.2, 4.2) and supervised (Section 4.2) formulas.

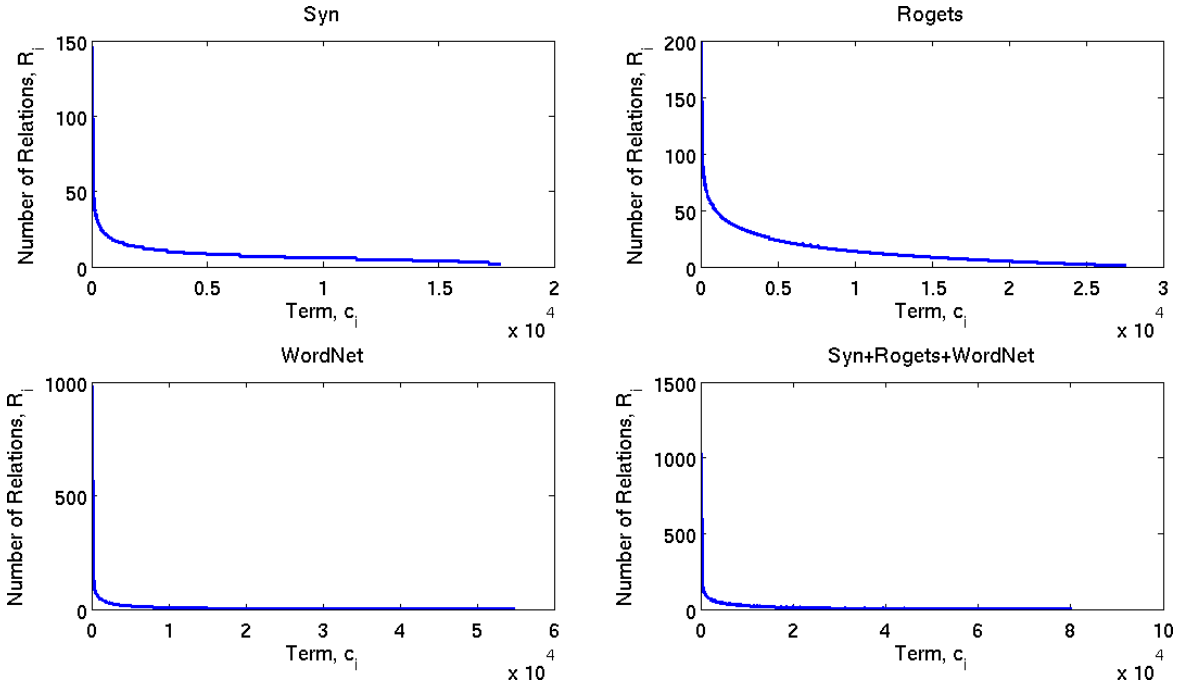


Figure 1.14: Number of relations (synonyms and hyponyms) per term in the dictionaries: a dictionary of synonyms, Roget's thesaurus, WordNet and a union of these three resources.

Semantic Similarity Measures

A *semantic similarity measure* is a specific similarity measure designed to quantify semantic relatedness of lexical units (here nouns and multiword expressions):

$$s_{ij} = sim(c_i, c_j), \text{ where } c_i, c_j \in C \text{ and } s_{ij} \in [0; 1]. \tag{1.23}$$

It yields high values for the pairs $\langle c_i, c_j \rangle$ in a semantic relation R (synonyms, hyponyms or co-hyponyms) and zero values for all other pairs:

$$s_{ij} = \begin{cases} \text{high} & \text{if } \langle c_i, c_j \rangle \in R, \\ 0 & \text{otherwise.} \end{cases} \tag{1.24}$$

A semantic similarity measure has the same properties as the similarity measure: positivity, symmetry and identity. Normally, each term in a language $c \in C$ has only few (like 5 or 50) semantically related words and many (like 200,000) unrelated words. Figure 1.14 illustrates this with a dictionary of synonyms [42], Rogets thesaurus (Kennedy and Szpakowicz, 2008) and WordNet (Fellbaum, 1998). Therefore, similarity scores of an adequate measure are often distributed according to a power law (see Figure 1.15). In the next chapters, for brevity, we will use the terms "similarity measure" and "measure" as an alias of "semantic similarity measure".

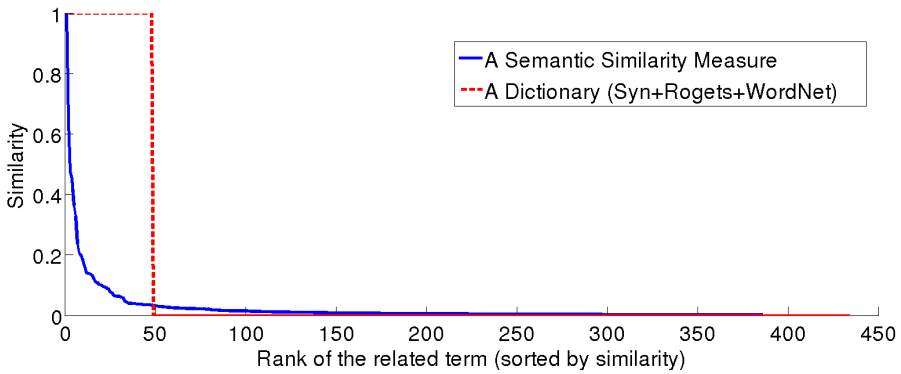


Figure 1.15: Similarity distribution of the term "doctor" in three dictionaries and as extracted by a semantic similarity measure. The total number of words (the X axis) is greater than 200,000.

Nearest Neighbor Procedures

In this thesis, we use several variations of the k-NN procedure to extract semantic relations: k-NN, mk-NN, p-NN and pk-NN. Each procedure takes as an input a sparse $|C| \times |C|$ similarity matrix $\mathbf{S}$, where each element s_{ij} represents a semantic similarity of terms $c_i, c_j \in$

[42] Synonyms database, `http://synonyms-database.downloadaces.com/`

C. The output of the procedure is a set of binary relations: $\hat{R} \in C \times C$.

- k-NN. This is the standard procedure, which links each term c_i with its k most similar neighbors, according to the scores provided in $\mathbf{S}$:
$$\hat{R} = \bigcup_{i=1}^{|C|} \{\langle c_i, c_j \rangle : (c_j \in top(k, c_i)) \wedge (s_{ij} > \gamma)\}, \tag{1.25}$$
where $k \geq 1$ is a desired number of relations per term, $\gamma \in [0; 1]$ is a global similarity threshold, which usually equals zero or a small value. Function top returns a set of the k most similar terms of a given term c_i:
$$top(k, c_i) = \{c_j : (s_{ij} \geq 0) \wedge (\sum_{\forall k: s_{ij} \leq s_{ik}} 1 \leq k)\}. \tag{1.26}$$
- mk-NN. This procedure keeps only mutual nearest neighbors within k most similar terms of c_i:
$$\hat{R} = \bigcup_{i=1}^{|C|} \{\langle c_i, c_j \rangle : (c_j \in top(k, c_i)) \wedge (c_i \in top(k, c_j)) \wedge (s_{ij} > \gamma)\}. \tag{1.27}$$
- p-NN. This procedure solely relies on the global similarity threshold γ. Let the similarity matrix $\mathbf{S}$ have $N \leq |C|^2$ non-zero elements. Then, the relations are established among the $\lfloor N - \frac{p}{100} N \rfloor$ most similar pairs $\langle c_i, c_j \rangle$, ranked by their similarity score $s_{ij} \in \mathbf{S}$. Here $p \in [0; 100]$ defines a percent of pairs which will be stripped off. The global similarity threshold γ equals the p-th percentile:
$$\hat{R} = \bigcup_{i=1}^{|C|} \{\langle c_i, c_j \rangle : s_{ij} > percentile(p)\}. \tag{1.28}$$
Figure 1.16 illustrates the result of the p-NN procedure for $p \in \{1, 5, 10\}$.
- pk-NN. This procedure combines local thresholding (as in k-NN) and global thresholding (as in p-NN):
$$\hat{R} = \bigcup_{i=1}^{|C|} \{\langle c_i, c_j \rangle : (c_j \in top(k, c_i)) \wedge (s_{ij} > percentile(p))\}. \tag{1.29}$$

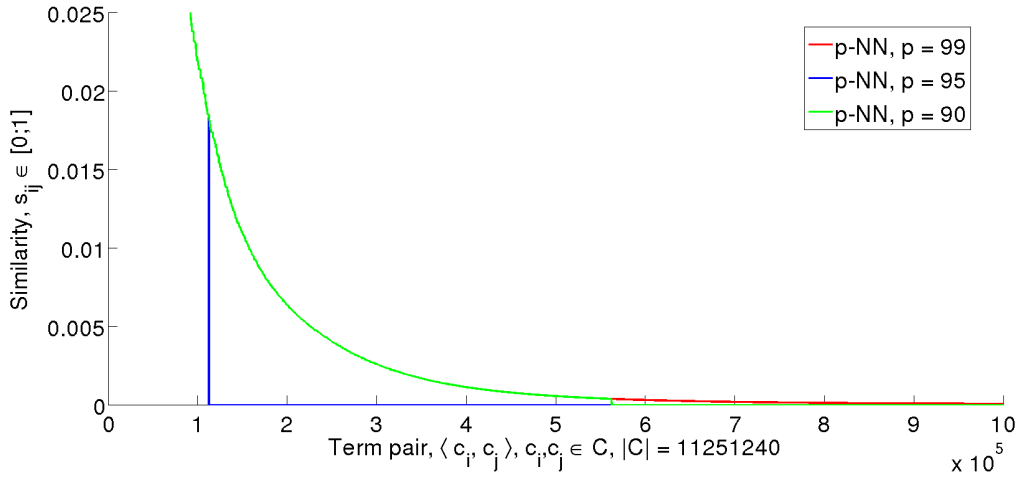


Figure 1.16: Result of the p-NN procedure.

1.2.3 Evaluation

There are various ways to evaluate and compare performances of semantic similarity measures, each with its pros and cons. In this thesis, we are going to evaluate them in the context of four tasks: [43]

1. correlations with human judgments (Chapters 3 and 4 and Sections 2.2 and 5.1);
2. semantic relation ranking (Chapters 3 and 4 and Sections 2.2 and 5.1);
3. semantic relation extraction (Sections, 2.2, 2.3, 2.4, 5.1);
4. using extracted relations in a text processing application:
   - in a context of short text classification system (Section 5.2);
   - in a context of a lexico-semantic search engine (Section 5.1).

The first three tasks are *intrinsic evaluations* as they characterize performance of a method with respect to a golden standard (Jones and Galliers, 1995). If a method can extract relations and similarities from the golden standard, then it is considered to be useful. The intrinsic evaluations quantify similarity of the extracted and handcrafted relations. Let a ground truth contain the synonymy relation $\langle car, vehicle \rangle$. Then, a method which is able to extract this relation will score higher than those which cannot do so. An intrinsic evaluation relies on a dataset O containing a desired output of a method. Performance of a method is a function of the real output O' and the desired output O: $f(O, O') \rightarrow \mathbb{R}$. This kind of evaluations are easily repeatable. Once O and f are given, it is straightforward to compute performance of a new method. Then, performance of two methods with outputs equals O' and O'' respectively can be compared as following: $f(O, O') - f(O, O'')$. However,

[43] Datasets and scripts used to perform evaluations on the first three tasks are available at: `https://github.com/alexanderpanchenko/sim-eval/`

choice of an appropriate golden standard O is not straightforward for the semantic relation extraction task. Curran (2003) underlines that the intrinsic evaluations tend to provide many false negatives. If an extracted relation $\langle car, ambulance \rangle$ is not in the golden standard O, then it will be considered as a wrong extraction.

The last two tasks are *extrinsic evaluations* as they characterize performance of a method in a context of an application. In this case, results of the extraction are used by an NLP system. If the extracted relations improve an overall performance of the system, then the extraction method is considered to be useful. This kind of evaluation shows us if the extracted relations are useful for a particular application. We can interpret a positive result as confirmation of the plausibility of the extracted semantic relations. Performance of the system is evaluated with respect to a golden standard O_s. Performance of a method is thus equal to the performance of the system: $g(O_s, O'_s) \rightarrow \mathbb{R}$. Here O'_s is an output of the system. This kind of evaluation requires more work as (a) the extracted relations should be integrated into a baseline system; (b) the extracted relations can be integrated into a system in many ways. That is why, it is more difficult to compare methods with this approach. On the other hand, application-based evaluations are considered to be important as they show if the extracted knowledge improves a real language processing system.

Correlation with Human Judgments

This kind of evaluation is a standard way to assess a semantic similarity measure. We rely on three classical human judgment datasets: MC (Miller and Charles, 1991), RG (Rubenstein and Goodenough, 1965) and WordSim (Finkelstein et al., 2001). These golden standards are widely used in the literature on semantic similarity. For example, they were used by Resnik (1995), Jiang and Conrath (1997), Lin (1998a), Budanitsky and Hirst (2006), Patwardhan and Pedersen (2006), Strube and Ponzetto (2006), Bollegala et al. (2007), Hughes and Ramage (2007), Zesch et al. (2008b), Agirre et al. (2009) and Yeh et al. (2009).

MC, RG and WordSim contain 30, 65 and 365 pairs of terms respectively. Each dataset is composed of N tuples $\langle c_i, c_j, s_k \rangle$, where c_i, c_j are terms and $s_k = s_{ij}$ is their similarity obtained by judgment of several native English speakers. Let $\mathbf{s} = (s_1, s_2, \ldots, s_N)$ be a vector of ground truth scores, and $\hat{\mathbf{s}} = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_N)$ be a vector of similarity scores calculated by a measure. Then, the quality of the measure is assessed with Pearson correlation coefficient r and Spearman's rank correlation coefficient ρ between $\mathbf{s}$ and $\hat{\mathbf{s}}$ (Howell, 2010):

$$r = \frac{cov(\mathbf{s}, \hat{\mathbf{s}})}{\sigma_{\mathbf{s}} \sigma_{\hat{\mathbf{s}}}} \approx \frac{\sum_{i=1}^{N} (s_i - \bar{s})(\hat{s}_i - \hat{\bar{s}})}{\sqrt{\sum_{i=1}^{N} (s_i - \bar{s})^2} \sqrt{\sum_{i=1}^{N} (\hat{s}_i - \hat{\bar{s}})^2}}, \bar{s} = \frac{1}{N} \sum_{i=1}^{N} s_i, \hat{\bar{s}} = \frac{1}{N} \sum_{i=1}^{N} \hat{s}_i, \quad (1.30)$$

where σ_s is the standard deviation of vector $\mathbf{s}$:

$$\sigma_s \approx \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(s_i - \hat{s})^2}, \tag{1.31}$$

the standard deviation of vector $\hat{\mathbf{s}}$ is defined in the same way and $cov(\mathbf{s}, \hat{\mathbf{s}})$ is a sample covariance of vectors $\mathbf{s}$ and $\hat{\mathbf{s}}$:

$$cov(\mathbf{s}, \hat{\mathbf{s}}) = \frac{\sum_{i=1}^{N}(s_i - \bar{s})(\hat{s}_i - \hat{\bar{s}})}{N-1}. \tag{1.32}$$

Spearman's rank correlation coefficient ρ is a Pearson correlation coefficient calculated on ranks rather than on absolute values:

$$\rho = \frac{cov(\mathbf{r}, \hat{\mathbf{r}})}{\sigma_{\mathbf{r}}\sigma_{\hat{\mathbf{r}}}} = \frac{\sum_{i=1}^{N}(r_i - \bar{r})(\hat{r}_i - \hat{\bar{r}})}{\sqrt{\sum_{i=1}^{N}(r_i - \bar{r})^2}\sqrt{\sum_{i=1}^{N}(\hat{r}_i - \hat{\bar{r}})^2}}. \tag{1.33}$$

Here r_i is the rank of the tuple $\langle c_i, c_j, s_k \rangle$ among other tuples if the dataset is sorted by s_k (see Table 1.4). Both correlation coefficients range from -1 to +1. Resnik (1995) replicated an experiment of Miller and Charles and reported a Pearson correlation of 0.902 between the original scores and his result. Jiang and Conrath (1997) obtained a correlation of 0.884 in a similar replication experiment. Because of this inherent subjectivity of the task, evaluations based on these datasets rarely achieve correlations higher than 0.9. One may interpret this, as a "subjectivity error" of around 0.1 (5%). Such errors are present in all evaluations used in this thesis, as each of them includes a subject "in the loop".

Table 1.4 depicts the MC dataset with scores obtained by a similarity measure. Figure 1.17 visualizes a Spearman's rank correlation between these human judgments and scores provided by a semantic similarity measure. As this evaluation is widely used by the community, the results may be compared with other published methods. However, the MC, RG and WordSim have a small vocabularies. Imagine a measure which performs well only on the 39 words of the MC dataset. Then, it may achieve the same correlation as a measure with coverage of 390.000 words.

Main characteristics of this evaluation task are the following:

- $(+)$ These datasets are widely used. Numerous publications during the last 20 years, make it easy to compare the results with the baselines.
- $(-)$ Small vocabulary size makes it impossible to assess coverage or recall of a similarity measure.
- $(-)$ The datasets provide no relation types.

The next evaluation task addresses disadvantages of this evaluation method.

Table 1.4: Miller-Charles (MC) dataset and scores obtained with a similarity measure.

Word, c_i	**Word**, c_j	**Human Score**, s_k	**Score**, $\hat{s}_k$	**Human Rank**, r_k	**Rank**, $\hat{r}_k$
automobile	car	3.92	0.884	1	1
journey	voyage	3.84	0.592	2	8
gem	jewel	3.84	0.581	3	3
boy	lad	3.76	0.325	4	2
coast	shore	3.70	0.440	5	7
asylum	madhouse	3.61	0.190	6	5
magician	wizard	3.50	0.556	7	4
midday	noon	3.42	0.692	8	10
furnace	stove	3.11	0.296	9	9
food	fruit	3.08	0.300	10	13
bird	cock	3.05	0.145	11	16
bird	crane	2.97	0.190	12	12
implement	tool	2.95	0.260	13	6
brother	monk	2.82	0.174	14	21
crane	implement	1.68	0.016	15	14
brother	lad	1.66	0.219	16	11
car	journey	1.16	0.124	17	25
monk	oracle	1.10	0.057	18	17
cemetery	woodland	0.95	0.056	19	24
food	rooster	0.89	0.027	20	26
coast	hill	0.87	0.186	21	28
forest	graveyard	0.84	0.069	22	23
shore	woodland	0.63	0.076	23	22
monk	slave	0.55	0.101	24	18
coast	forest	0.42	0.145	25	19
lad	wizard	0.42	0.083	26	20
cord	smile	0.13	0.020	27	29
glass	magician	0.11	0.078	28	27
noon	string	0.08	0.026	29	15
rooster	voyage	0.08	0.005	30	30

Semantic Relation Ranking

This evaluation task stems from the work of Baroni and Lenci (2011) on the BLESS dataset (Baroni and Lenci Evaluation of Semantic Spaces). This benchmark uses a set of semantic relations R, such as $\langle agitator, syn, activist\rangle$, $\langle hawk, hyper, predator\rangle$, $\langle gun, syn, weapon\rangle$, $\langle dishwasher, cohypo, freezer\rangle$, $\langle lecture, random, clown\rangle$, $\langle driver, random, stone\rangle$, and $\langle computer, random, river\rangle$. Each "target" term has roughly the same number of meaningful and random relations. A measure should rank semantically similar pairs higher than the random ones.

We use two semantic relation datasets: BLESS (Baroni and Lenci, 2011) and SN (Panchenko and Morozova, 2012). BLESS relates 200 target nouns to 8,625 relatums with 26,554 se-

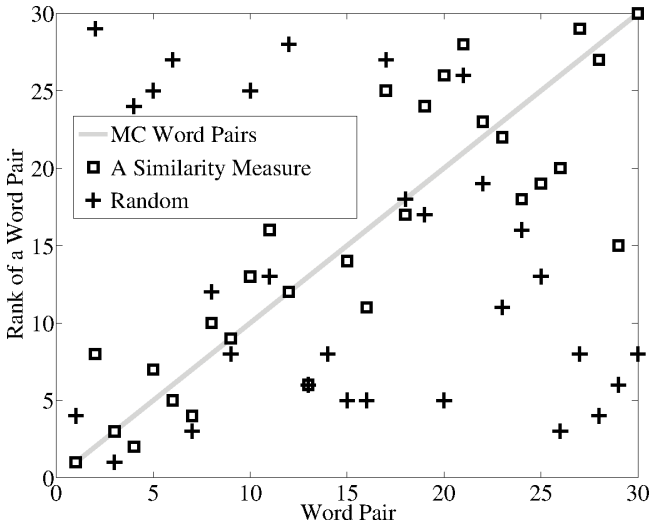


Figure 1.17: Spearman's rank correlation between human judgments from Miller-Charles (MC) word pairs and scores provided by a semantic similarity measure. Spearnam's correlation ρ of a similarity measure equals 0.843 (p<0.001) and a correlation of the random measure equals -0.173 (p=0.360).

mantic relations (14,440 relations are meaningful and 12,154 relations are random) [44]. Every relation has one of the following types: hypernymy, co-hyponymy, meronymy, attribute, event or random. BLESS contains semantic relations from the McRae Norms (McRae et al., 2005), WordNet (Miller, 1995a), ConceptNet (Liu and Singh, 2004) and text corpora. The relations were validated manually and through crowd-sourcing.

We built the SN (Semantic Neighbors) dataset in order to complement the BLESS as it contains no synonyms. SN relates 462 target nouns to 5,910 relatum words with 14,682 relations (7,341 synonyms and 7,341 are random) [45]. The SN contains synonyms coming from three sources: WordNet 3.0 (Miller, 1995b), Roget's thesaurus (Kennedy and Szpakowicz, 2008) and a synonyms database [46]. The relations in the dataset were validated manually.

In this task, a similarity measure is used to rank related words of each "target". Here a "target" term has roughly the same number of meaningful and random "relatums". Table 1.5 depicts 112 relations of the word "hawk" from the BLESS ranked by similarity score. The evaluation is based on the number of correct relations in the "top list". First, we rank the relations by target word c_i and similarity score $s_k = s_{ij}$ as in Table 1.5. Second, each term c_i is linked with $k\%$ of its nearest neighbors (a kind of k-NN):

$$\hat{R} = \bigcup_{i=1}^{|C|} \{\langle c_i, c_j \rangle : (c_j \in \text{ top } k\% \text{ terms of } c_i) \wedge (s_{ij} \geq 0)\}, s_{ij} \in \mathbf{S}. \tag{1.34}$$

[44] https://sites.google.com/site/geometricalmodels/shared-evaluation
[45] https://github.com/alexanderpanchenko/sn
[46] http://synonyms-database.downloadaces.com/

Table 1.5: A target word "hawk" and all its relatum words from the BLESS dataset ranked by similarity score. The table on the left contains relations retrieved with the k-NN threshold of 50% The whole table contains all relations of the word ($k = 100\%$).

Target, c_i	**Relatum**, c_j	**Relation Type**, t	**Score**, $\hat{s}_k$
hawk	bird	hyper	1
hawk	dove	cohypo	1
hawk	eagle	cohypo	1
hawk	falcon	cohypo	1
hawk	owl	cohypo	1
hawk	vulture	cohypo	1
hawk	sparrow	cohypo	0.99999
hawk	swoop	event	0.99997
hawk	raptor	hyper	0.99996
hawk	swan	cohypo	0.99987
hawk	nest	event	0.99976
hawk	woodpecker	cohypo	0.99973
hawk	feather	mero	0.99964
hawk	crow	cohypo	0.99952
hawk	pigeon	cohypo	0.99948
hawk	fly	event	0.99946
hawk	goose	cohypo	0.999
hawk	pheasant	cohypo	0.99868
hawk	predator	hyper	0.9985
hawk	wing	mero	0.99836
hawk	beak	mero	0.99824
hawk	soar	event	0.99733
hawk	robin	cohypo	0.99644
hawk	penguin	cohypo	0.99451
hawk	roost	event	0.99013
hawk	creature	hyper	0.97707
hawk	animal	hyper	0.97269
hawk	head	mero	0.97016
hawk	claw	mero	0.96007
hawk	gray	attri	0.96
hawk	hunt	event	0.95608
hawk	plumage	mero	0.95465
hawk	hover	event	0.93882
hawk	catch	event	0.92227
hawk	vertebrate	hyper	0.91701
hawk	eye	mero	0.88698
hawk	wild	attri	0.88417
hawk	grey	attri	0.8643
hawk	young	attri	0.79082
hawk	foot	mero	0.78725
hawk	plume	mero	0.76141
hawk	passenger	random	0.66433
hawk	spot	event	0.65185
hawk	sit	event	0.65024
hawk	big	attri	0.6362
hawk	circle	event	0.58847
hawk	brown	attri	0.55922
hawk	chordate	hyper	0.54453
hawk	other	random	0.54346
hawk	sea	random	0.53117
hawk	windmill	random	0.51327
hawk	see	event	0.51181
hawk	first	random	0.49451
hawk	eat	event	0.45854
hawk	strong	attri	0.42262
hawk	aggressive	attri	0.34521

Target, c_i	**Relatum**, c_j	**Relation Type**, t	**Score**, $\hat{s}_k$
hawk	reverse	random	0.34406
hawk	include	random	0.32461
hawk	large	attri	0.29802
hawk	further	random	0.29732
hawk	star	random	0.27972
hawk	prism	random	0.2759
hawk	talon	mero	0.27262
hawk	rump	mero	0.2581
hawk	dark	random	0.24776
hawk	aim	random	0.24598
hawk	lay	event	0.24087
hawk	paint	random	0.23446
hawk	old	attri	0.22896
hawk	pinion	mero	0.21244
hawk	experienced	random	0.20552
hawk	crate	random	0.20478
hawk	live	event	0.19195
hawk	inhabit	event	0.18192
hawk	christmas	random	0.18135
hawk	conference	random	0.16068
hawk	trim	random	0.15775
hawk	limestone	random	0.14231
hawk	breathe	event	0.13926
hawk	die	event	0.13781
hawk	binding	random	0.11747
hawk	convict	random	0.11744
hawk	sign	random	0.11432
hawk	kerb	random	0.11242
hawk	nominate	random	0.11112
hawk	jacaranda	random	0.10915
hawk	strengthen	random	0.10065
hawk	everytime	random	0.090728
hawk	present	random	0.088429
hawk	difficulty	random	0.080802
hawk	elapse	random	0.08037
hawk	practical	random	0.07223
hawk	no-one	random	0.071844
hawk	educational	random	0.06702
hawk	economic	random	0.065984
hawk	judicial	random	0.065793
hawk	concern	random	0.065013
hawk	contextual	random	0.064861
hawk	feign	random	0.063987
hawk	localise	random	0.063514
hawk	neutron	random	0.06142
hawk	msm	random	0.060879
hawk	genesis	random	0.060742
hawk	improvisation	random	0.059412
hawk	employer	random	0.059407
hawk	triathlon	random	0.056394
hawk	idealism	random	0.055417
hawk	fia	random	0.054996
hawk	co-operative	random	0.047968
hawk	ira	random	0.044595
hawk	mindfulness	random	0.042337
hawk	shortcoming	random	0.031764

Let $\hat{R}_k$ be a set containing top k % semantic relations for each target word c_i and R be a set of all correct (i. e., non-random) semantic relations. Then, Precision, Recall, F1-measure are calculated as follows:

$$Precision(k) = \frac{|R \cap \hat{R}_k|}{|\hat{R}_k|}, Recall(k) = \frac{|R \cap \hat{R}_k|}{|R|}, \tag{1.35}$$

$$Fmeasure(k) = \frac{Precision(k) \cdot Recall(k)}{Precision(k) + Recall(k)} \tag{1.36}$$

Each "target" term c_i has roughly the same number of meaningful and random relations. Therefore, random measure approximately equals 0.5 and not 0 as in the case of the open vocabulary relation extraction (see below). Table 1.5 illustrates a threshold value k of 50%. Here the table on the left contains all relations retrieved for this threshold value, while table on the right contains all other relations.

This benchmark quantifies relative performances of the measures. These relative scores should not be confused with the absolute scores of the open-vocabulary relation extraction (see the next section). In the current task, the quality of a similarity measure is assessed with six criteria. Each of them brings a specific kind of information about the measure:

1. $Precision(10)$ is the precision of the 10% top results. This statistic indicates if the measure can extract at least several related words. $Precision(10)$ is a good criterion if an application which uses the measure prioritizes precision over recall.

2. $Precision(20)$ is the precision of the first 20% results. This statistic is similar to the previous one. It is a good criterion for the application which favors precision over recall and at the same time is relatively robust to the noisy results.

3. $Precision(50)$ is the precision of the top 50% results. If $Precision(50) = 1$ and the test dataset if balanced, then the similarity measure is optimal. It puts on the top 50% semantically related pairs and on the bottom the rest 50% random pairs.

4. $Recall(50)$ is the recall of the first 50% results. It equals $Precision(50)$ if the number of extracted relations $|\hat{R}_{50}|$ equals the number of known relations $|R|$:
$$|\hat{R}_{50}| = |R| \Rightarrow \frac{|\hat{R}_{50} \cap R|}{|\hat{R}_{50}|} = \frac{|\hat{R}_{50} \cap R|}{|R|} \Rightarrow Precision(50) = Recall(50). \tag{1.37}$$
However, in our evaluation framework, we do not count the extractions with zero similarity scores: $\forall \langle c_i, c_j \rangle \in \hat{R} : s_{ij} \neq 0$. Thus, for some sparse measures, $Precision(50) \neq Recall(50)$.

5. $Fmeasure(50)$ is the harmonic mean of $Precision(50)$ and $Recall(50)$. It is a good criterion for an application that requires both precision and recall and is robust to the noisy results.

6. *Precision-Recall plot* takes the whole precision-recall curve into account and is threshold independent (see Figure 1.18). Thus, such plot summarizes performance of a similarity measure across different levels of the k threshold. Figure 1.18 shows that the 100% recall corresponds to the precision of 55%. It is so because the test dataset contains 55% of semantic relations and 45% of random pairs. Precision-Recall graphs are useful for analysis and comparison of the measures.

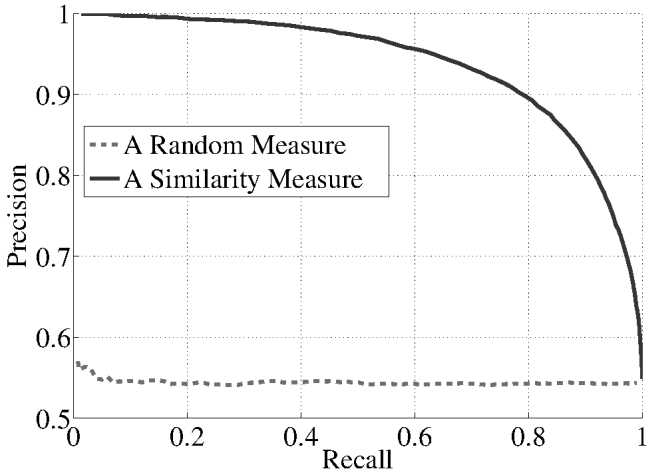


Figure 1.18: A precision-recall graph: performance of a similarity measure on the semantic relation ranking task across different values of the threshold k.

Semantic relation ranking is not as wide-spread in the literature as correlations with human judgments. So far, BLESS was used as a golden standard in the following works: (Baroni and Lenci, 2011), (Lenci and Benotto, 2012), (Panchenko, 2011), (Panchenko, 2012) and (Panchenko and Morozova, 2012). SN was used as a golden standard in (Panchenko, 2012), (Panchenko and Morozova, 2012), and (Panchenko et al., 2012). However, this kind of evaluation is interesting because of its size. The vocabulary of MC, RG and WordSim consists of 39, 48 and 437 words, respectively. The vocabulary of BLESS and SN contains 8,026 and 5,910 words, respectively. A big vocabulary and a big number of semantic relations (26,554 and 14,682 relations, respectively) let us estimate recall of a similarity measure.

Main characteristics of this evaluation task are the following:

- $(+)$ These datasets are two order of magnitude larger than those of human judgements. This let us estimate recall and lexical coverage of a similarity measure.
- $(+)$ The datasets list several relation types between terms: hypernyms, co-hyponyms, etc. This let us analyse performances with respect to different relation types.
- $(-)$ The datasets were introduced recently. There are only few studies which used it.
- $(-)$ This task does not let us estimate the real performance of the semantic relation extraction. It rather provides relative performances of the measures.

The next evaluation task addresses the disadvantages of this evaluation method.

Semantic Relation Extraction

An output of a semantic relation extraction procedure is a set of relations $\hat{R}$. Each "target" term $c_i \in C$ has a set of related words $C_i \subset C : \forall c \in C_i \exists \langle c_i, c \rangle \in \hat{R}$. In this task, the quality of a similarity measure is assessed with the precision of the extracted relations $\hat{R}$ against a golden standard set of relations R:

$$Precision = \frac{|R \cap \hat{R}|}{|\hat{R}|}. \tag{1.38}$$

Thus, precision is a fraction of correct relations among all extracted relations according to a golden standard R.

It is also possible to calculate recall and F1-score based on R:

$$Recall = \frac{|\hat{R} \cap R|}{|R|}, Fmeasure = \frac{Precision \cdot Recall}{Precision + Recall}. \tag{1.39}$$

This evaluation protocol is a standard way to benchmark semantic relation extraction. Originally proposed by Grefenstette (1994), a similar assessment methodology was adapted by various researchers. In his original work, Grefenstette used as a golden standard R a combination of the Roget's 1911 thesaurus, the Macquarie Encyclopedic Thesaurus and the Webster's 7th Edition dictionary. Later, researchers conducted evaluations based on golden standards. Curran (2002, 2003) and Curran and Moens (2002) compiled a golden standard from Roget's 1911 (Roget, 1911), Roget's II (Hickok, 1995) thesauri, Moby Thesaurus (Ward, 1996), The New Oxford Thesaurus of English (Hanks, 2000), and The Macquarie Encyclopedic Thesaurus (Bernard, 1990). Chen (2006) used an astronomy thesaurus. Sahlgren (2006) used Moby Thesaurus and University of South Florida Association Norms (Nelson et al., 2004). Morlane-Hondère and Fabre (2012) used a large database of free associations "JeuxDeMots" for evaluation of a distributional similarity measure. Sang and Hofmann (2007, 2009) evaluated hypernym extraction with the EuroWordNet (Ellman, 2003). Takenobu et al. (1995) used a Japanese thesaurus to evaluate quality of an automatic thesaurus construction system.

A golden standard R may be based on a dictionary, as in the papers mentioned above, or alternatively it can be obtained by human judgment as in (Pantel et al., 2004), (Rybinski et al., 2007), (Asuka et al., 2008) or (Panchenko et al., 2012). In this case, a small random sample of the extracted relations is annotated by humans. The result of these annotations is

a ground truth R which precisely corresponds to a set of extracted relations: $R \subseteq \hat{R}$. The agreement between annotators is calculated with Fleiss' kappa statistic (Fleiss, 1971).

Main characteristics of this evaluation task are the following:

- $(+)$ It lets us estimate the real relation extraction precision. This metric is usually comparable to the previously published results.
- $(-)$ If a golden standard is based on a dictionary, then there is a high chance of false positives.
- $(-)$ If a golden standard is based on an annotation of the extracted relations, then the evaluation becomes hardly reproducible. Furthermore, this benchmark requires much annotation effort and thus usually performed on a small sample.

Difference between Semantic Relation Ranking and Extraction

In semantic relation extraction, a term $c_i \in C$ may be related to any other term from the vocabulary C: $\hat{R} \subset C \times C$. On the other hand, in semantic relation ranking, a term c_i may be related only to a small predefined number (like 100) of terms from the vocabulary C: $\hat{R} = \bigcup_{i=1}^{|C|} \hat{R}_i$, where $\hat{R}_i \subset C \times C_i$ is a set of relations with the target word c_i: $|C_i| \ll |C|$.

Precision of extraction with a random similarity measure equals the probability that a randomly generated relation $\hat{r} \in \hat{R}_{rand}$ is correct:

$$Precision_{random} = P(\hat{r} \in R). \tag{1.40}$$

For the semantic relation extraction task, precision of a random measure equals zero due to the exponential distribution of the relations (see Figures 1.15 and 1.14):

$$Precision_{random} = \frac{1}{|C|^2} \approx 0 \tag{1.41}$$

On the other hand, for the semantic relation ranking task based on a balanced dataset, precision of a random measure equals 0.5:

$$Precision_{random} = \frac{|R_{syn} \cup R_{cohypo} \cup R_{attri} \cup R_{event} \cup R_{mero}|}{|R_{random}|} \approx 0.5. \tag{1.42}$$

1.3 Conclusion

First, in this chapter, we have introduced the context of the research. Semantic resources, such as thesauri or ontologies are useful for various text processing and information retrieval applications. These resources are composed of semantic relations. Second, we motivated

and formulated the research question. Due to the limitations of the manually-constructed resources and sub-optimal performance of the existing relation extractors, novel approaches to semantic relation extraction are needed. Third, we have presented a key steps of our extraction framework. Its main components are a semantic similarity measure and a nearest neighbor procedure. Finally, we have introduced a set of evaluation tasks designed to benchmark such similarity-based semantic relation extractors.

Chapter 2

Single Semantic Similarity Measures

Always try the simple solution first.

– Unknown author.

This chapter deals with *single* semantic similarity measures. These measures rely on one source of information (a corpus, a dictionary, etc.) and on one extraction method (distributional analysis, lexico-syntactic patterns, etc.). Section 2.1 begins with an overview of the existing single semantic similarity measures. Next, we present three novel semantic similarity measures. Section 2.2 presents a similarity measure based on syntactic distributional analysis. We started the work with the distributional measures as they derive similarity scores from a text corpus in an unsupervised data-driven manner. At the next stage of our research for a quality semantic similarity measure, we looked at the dictionary-based approaches. Section 2.3 presents a new similarity measure based on definitions from Wiktionary and Wikipedia. Our experiments with distributional and dictionary-based measures revealed the following. The corpus-based approach provides a good lexical coverage, but the simple distributional representation sometimes hampers precision. On the other hand, manually-crafted definitions provide a precise estimation of the term similarity. However, definitions provide much smaller coverage in comparison with the corpus-based methods. Therefore, we present in Section 2.4 an original similarity measure which extracts "definitions" from a huge corpus with lexico-syntactic patterns. The extracted contexts are then used to estimate semantic similarity.

2.1 Related Work

There exists a significant body of literature on semantic similarity measures. Most current approaches use a *single* source of information to derive a similarity score between words.

The well-established sources of information are corpora, semantic networks, dictionaries and Web as a corpora.

Measures based on the WordNet semantic network (Miller, 1995b) were proposed by Wu and Palmer (1994), Resnik (1995), Jiang and Conrath (1997), Leacock and Chodorow (1998) and Lin (1998a). Other network-based measures were proposed by Gurevych (2005), Zesch and Gurevych (2007) and Kennedy and Szpakowicz (2008). Measures of semantic similarity based on dictionaries and the Vector Space Model were proposed by Lesk (1986), Fox et al. (1988), Zesch et al. (2007) and Zesch et al. (2008b). Blondel and Senellart (2002), Ho and Fairon (2004), Muller et al. (2006) and Navarro et al. (2009) proposed similarity measures based on definitions and graph-based models. Several recent successful dictionary-based approaches that rely on Wiktionary and/or Wikipedia: Navarro et al. (2009), Zesch et al. (2007) or Zesch et al. (2008b). Corpus-based methods of semantic relation extraction based on lexical and dependency patterns were proposed by Hearst (1992), Snow et al. (2004), Bollegala et al. (2007) and Sang and Hofmann (2009). Auger and Barrière (2008) performed a comprehensive survey of the pattern-based methods.

Yet another well-known group of similarity measures is based on the Haris's Distributional Hypothesis which states that "words that occur in the same contexts tend to have similar meanings" (Harris, 1954). Schütze (1993) was the first to represent a word as a vector in a multidimensional space of its context in a corpus. The meaning of a word in this vector space is modeled by the spatial proximity of words. In the simplest case, the distributional analysis relies on the context window approach. However, there exist many variations of this technique including those proposed by Crouch and Yang (1992), Takenobu et al. (1995), Philippovich and Prokhorov (2002), Van Der Plas and Bouma (2004) and Martin and Azmi-Murad (2005). These traditional distributional models were extended with clustering algorithms by Crouch (1988), Pereira et al. (1993), Lin and Pantel (2001), Caraballo (2001) and Pantel et al. (2004). A vector-space model based on syntactic contexts, the so called Syntactic Distributional Analysis (*SDA*), was proposed by Grefenstette (1994) and further developed by Lin (1998b), Padó and Lapata (2007) and Erk and Padó (2008). The key findings on distributional analysis are summarized by Curran (2003) and Bullinaria and Levy (2007). Other successful corpus-based approaches to lexical semantics based on the Vector Space Model include Hyper Analogue to Language (*HAL*) (Lund and Burgess, 1996), Random Projection (*RP*) (Bingham and Mannila, 2001) and Reflective Random Indexing (Cohen et al., 2010).

Latent Semantic Analysis (*LSA*) (Landauer and Dumais, 1997) and topic models such as Probabilistic Latent Semantic Analysis (*PLSA*) (Hofmann, 2001) and Latent Dirichlet Allocation (Blei et al., 2003) are used to derive key topics from a collection of text documents (Hall et al., 2008). Each topic is distribution over the vocabulary of this collection.

Sometimes such topics are represented with several most probable terms, e. g. (Hall et al., 2008):

- *Information Retrieval:* document, documents, query, retrieval, question, information, answer, term, text, web, ...
- *Lexical Semantics:* semantic, relations, domain, noun, corpus, relation, nouns, lexical, ontology, patterns, ...
- *Information Extraction:* system, text, information, muc, extraction, template, names, patterns, pattern, domain, ...

Here the names of the topics, such as "Information Retrieval", were introduced manually. Thus, each topic contains a set of topically-related terms. Such topical representations proven to be useful in several applications such as information retrieval, text categorization and text clustering (Lu et al., 2011).

It is possible to use the topic models and the related dimensionality reduction techniques to model semantic similarity. One way to do so is to assume that the most probable terms of a topic are semantically related. Griffiths et al. (2003) compares performance of the LSA and a topic model on the word association prediction task. According to these experiments, the topic model significantly outperforms the LSA in terms of correlations with human judgement. In (Griffiths et al., 2007), the author further develops the study of topic models applied to semantic representations. He shows that the topic models predict word associations more precisely than the LSA. Séaghdha (2010) uses three topic models based on the LDA to induce selectional preferences of terms. The author finds that performance of the topic models is competitive or superior to the baselines such as the web-based measures (see below).

Measures of semantic similarity which exploit the web as a corpus (Kilgarriff and Grefenstette, 2003) include Pointwise Mutual Information Information Retrieval (*PMIIR*) (Turney, 2001), Normalized Google Distance (*NGD*) (Cilibrasi and Vitanyi, 2007), *WebJaccard, WebDice, WebOverlap* (Bollegala et al., 2007) and *VGEM* (Veksler et al., 2008). Lindsey et al. (2007) performed a comprehensive study of web-based measures, testing various formulae, search engines and search domains. The authors came to the conclusion that a small search domain is better than the whole Web.

Recently, several prominent approaches based on Wikipedia were proposed. The *WikiRelate!* system designed by Strube and Ponzetto (2006) exploits the abstracts of articles and the network from Wikipedia categories. Gabrilovich and Markovitch (2007) and Zesch et al. (2008a) proposed alternative measures of semantic similarity based on texts of Wikipedia. These measures represent the concepts in a vector space of all Wikipedia articles. Nakayama et al. (2007) suggested yet another relation extraction method based on Wikipedia. The authors used the hyperlinks structure of Wikipedia articles to infer associations between words.

Finally, Milne et al. (2006) suggested to extract synonyms, hypernyms and associations from Wikipedia category lattice and other structure and navigational elements of Wikipedia.

Measures based on semantic networks such as Wu and Palmer (1994), Leacock and Chodorow (1998) and Resnik (1995) achieve high precision, but suffer from a limited coverage. Definition-based methods such as *ExtendedLesk* (Banerjee and Pedersen, 2003), *GlossVectors* (Patwardhan and Pedersen, 2006), have roughly the same properties as they rely on a manually-crafted semantic resource. On the other hand, corpus-based measures such as distributional analysis or *LSA*, provide acceptable recall as they can derive similarity score directly from a corpus. However, these methods suffer from lower precision as most of them rely on a simple representation based on the Vector Space Model.

2.2 SDA-MWE: A Similarity Measure Based on Syntactic Distributional Analysis[1]

In this section, we describe experiments with a similarity measure based on syntactic distributional analysis. We apply it to the semantic relation extraction task (see Section 1.2.3). Namely, we compare quality of the relations of a manually-crafted thesaurus with the relations extracted from text.

Thesauri proven to be useful for information retrieval and management (see Section 5.3). However, the traditional way to construct thesaurus involves a great amount of manual labor. One of the solutions to this problem is to automatize thesaurus construction, as it was proposed by Grefenstette (1994). The automatized process comprises two steps: selecting key terms for a given domain and establishing semantic relations between them. A key issue concerns the quality of an automatically generated thesaurus. In this section we focus on the second step.

Contributions of this section are two-fold:

- First, we present a study on automatic thesaurus construction with semantic similarity measures. In this experiment, we tried to automatically reconstruct relations of a thesaurus. We studied how similarity of the extracted relations and the relations established by an expert.
- Second, we proposed a similarity-based relation extraction method *SDA-MWE*, which stems from the syntactic distributional analysis. In contrast to the similar approaches, the method can deal with both single words and multiword expressions (MWEs).

We present the dataset used in the experiment in Section 2.2.1. Section 2.2.2 describes our method for semantic relations extraction. Then, in Section 2.2.3, we present our evaluation

[1]The research presented in this section has been published as Panchenko [11] and Панченко [14].

strategy. The results are reported in Section 2.2.4. Finally, we sum up the main points of the experiment in Section 2.2.5.

2.2.1 Dataset

The dataset consists of a 20 million word corpus of political texts in French and a manually constructed thesaurus [2]. The corpus encompasses 11,386 documents from a governmental institution, such as deputy requests to ministers, protocols of parliamentary sessions, international conventions, activity reports, texts of propositions of new laws and so on.

The thesaurus was constructed manually based on the analysis of the aforementioned corpus. The resource provides a vocabulary for indexing documents of a governmental institution such as a parliament. Thus, it includes terms from 12 domains often discussed in such an institution (legislation, economics, finances, international relations, etc.). The thesaurus contains $n = 2{,}514$ concepts $C = \{c_1, \ldots, c_n\}$, where every concept c_i is represented with j terms $\{d_{i1}, \ldots, d_{ij}\}$ which are synonyms or quasi-synonyms. For example, the concept "Aircraft" is composed of eight terms [3]:

$$c_i = \{d_{i1}, \ldots, d_{i8}\} = \{Aircraft, Airship, Plane, Aerostat, \ldots, Dirigible\}. \quad (2.1)$$

The vocabulary D of the thesaurus comprises $m = 4,771$ terms: $D = \bigcup_{c_i \in C} c_i = \{d_1, \ldots, d_m\}$. 65% of the terms in the vocabulary are noun phrases, such as "ultra-lightweight aircraft" or "hot-air balloon". The remaining 35% of nouns, like "airplane" or "aerostat". The concepts are hierarchically organized by means of 2,456 hypernymy relations R^{NT}, where NT stands for "Narrower Than". Furthermore, the concepts of the thesaurus are interconnected with the set of 1,530 associative relations R^{RT}, where RT stands for "Related To". Every semantic relation $r_{ij} \in \{R^{NT} \cup R^{RT}\}$ defines a semantic link between concepts c_i and c_j represented by the ordered pair $\langle c_i, c_j \rangle$. Thus, the thesaurus is the oriented graph (C, R) having the concepts of the thesaurus C as nodes and the semantic relations between concepts $R = R^{NT} \cup R^{RT}$ as edges.

2.2.2 Method

Given a corpus and a set of terms, the goal of the method is to establish relations between them. To achieve this goal, we adopt the syntactic distributional analysis, which models a term as a point in a space of all possible syntactic contexts (Grefenstette, 1993; Padó

[2] The dataset provided by an industrial partner of the STRATEGO research project (`http://cental.fltr.ucl.ac.be/projects/stratego/`). The thesaurus was handcrafted by the partner company.

[3] The examples are translated from French to English.

and Lapata, 2007). The method, called *SDA-MWE* (Syntactic Distributional Analysis for Multiword Expressions) involves four steps: preprocessing of the vocabulary and the corpus, indexing descriptors of the thesaurus, constructing a distributional space of descriptors and computing of relations between the descriptors.

Preprocessing of the Vocabulary and the Corpus

At this step, we perform a standard preprocessing: remove special characters and markup, normalize whitespaces, etc. We also substitute French diacritic symbols (e. g.,"à" or "é") with their non-diacritic equivalents (e. g., "a" or "e"), to take into account accent variation.

Descriptor Indexing

The goal of this step is to find all occurrences of the terms $d \in D$ in the corpus and save information about their positions. Indexing is necessary to link thesaurus descriptors with results of parsing (see below). An index record is a tuple $\langle d, doc, p^{beg}, p^{end} \rangle$, where p^{beg} and p^{end} are the positions of the beginning and the end of the term in the document doc. In order to deal with linguistic variation of descriptors, we generate a regular expression for each term. The procedure replaces every article or preposition with the disjunction of common articles and prepositions, e. g. `(a|aux|de|des|...|vers)`. Each noun, verb or adjective is replaced with the expression based on its stem.

For instance, the procedure transforms the descriptor "conventions internationales" into the expression which captures both singular form "convention internationale" and plural form "conventions internationals". Similarly, the expression for the term "modification de la legislation" captures different variations of the term, such as "modifications de la legislation", "modification a la legislation" or "modifications dans la legislation".

Constructing Distributional Space of Terms

To construct a distributional space, we use syntactic dependencies between words of sentences which contain at least one descriptor $d \in D$. To generate syntactic dependencies SR from the corpus, we used the natural language parser XIP (Aït-Mokhtar et al., 2002). Each dependency $\langle w_i, p_i^{beg}, t, w_j, p_j^{beg} \rangle$ represents a syntactic relation of the type t between the word w_i starting at the position p_i^{beg} and the word w_j starting at the position p_j^{beg}.

Some dependency types such as DET, APP and PREOBJ say nothing about the meaning of the head word (see Table 2.1). For instance, consider the following three dependencies:

- $\langle the, p_i, DET, plane, p_j \rangle$

- $\langle the, p'_i, DET, helicopter, p'_j \rangle$
- $\langle the, p''_i, DET, tomato, p''_j \rangle$

In this case, the dependency DET does not help us learn that "plane" is semantically more similar to "helicopter" than to "tomato". Thus, in our method, we use only nine syntactic dependencies listed in Table 2.1 [4]. The table also indicates syntactic relations used by other researchers. We can observe that the most popular relations are OBJ, SUBJ and ADJMOD.

Type	Description	A	B	C	D	E	F	G	H
ADJMOD	Attaches the modifier of adjective to the adjective itself.	X	X		X	X		X	X
CONNECT	Links the verb of a finite clause to the grammatical word that introduces the clause.	X				X	X		
COORD	Coordination. This binary relation links coordinated elements.	X	X			X		X	
DOBJ	This dependency attaches a deep object to the verb.	X				X	X		
DSUBJ	This dependency attaches a deep subject to the verb.	X				X	X		
NMOD	Attaches a modifier to the noun it modifies.	X				X			X
OBJ	Attaches a direct object to its verb.	X	X	X	X	X	X	X	X
SUBJ	Attaches a surface subject to the verb, including infinitive verbs.	X	X	X	X	X	X	X	X
VMOD	Attaches a modifier of a verb to the verb itself.	X				X	X		
DET	Links a nominal head and a determiner.				X	X			
APP	Apposition. Links two adjacent units that have identical referents.					X		X	X
PREPOBJ	Attaches a preposition to the noun or the verb it precedes.		X			X		X	X

Table 2.1: Syntactic relations used to construct distributional space by (A) our method, (B) Heylen et al. (2008), (C) Hindle (1990), (D) Hirschman et al. (1975), (E) Hatzivassiloglou and McKeown (1993), (F) Van Der Plas and Bouma (2004), (G) Takenobu et al. (1995), (H) Grefenstette (1994). We adapted descriptions from the documentation of the XIP parser (Aït-Mokhtar et al., 2002).

Thus, the dimensions of the distributional space must let us find semantically similar descriptors. In our approach, the dimensions of a n-dimensional space are associated with a set of *syntactic contexts* $B = \{\beta_1, \ldots, \beta_n\}$. Each syntactic context β is a tuple $\langle t, w \rangle$ composed of the lemmatized word w and the type of syntactic relation t. We derive the set of syntactic contexts from the set of extracted syntactic dependencies SR. A tuple $\langle w_i, p_i^{beg}, t, w_j, p_j^{beg} \rangle$ provides two syntactic contexts: $\langle t, w_i \rangle$ and $\langle t, w_j \rangle$. Each term d_i is represented with a vector $\mathbf{f}_i$ in the distributional space. The feature matrix $\mathbf{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_m)^T$ has m rows and n columns, the i-th row of the matrix corresponds to the term d_i and j-th column corresponds to the syntactic feature β_j.

We use Algorithm 1 to calculate the distributional space B and the feature matrix $\mathbf{F}$. Most

[4] Refer to `http://www.hutchinsweb.me.uk/IntroMT-2.pdf` for information about the difference between the surface subjects/objects (SUBJ/OBJ) and the deep subjects/objects (DSUBJ/DOBJ).

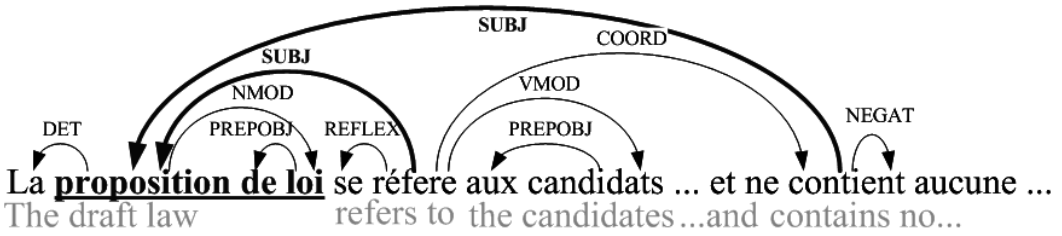


Figure 2.1: Representing the descriptor "proposition de loi" with syntactic features coming from the dependency parser XIP.

previous methods represent a single word or a chunk in the distributional space (compare, Hirschman et al. (1975), Hindle (1990), Hatzivassiloglou and McKeown (1993), Grefenstette (1994), Takenobu et al. (1995), Van Der Plas and Bouma (2004), Heylen et al. (2008)). The main difference of our method is that it can compute the distributional representation of an arbitrary multiword expression. It calculates the distributional representation of a term as a sum of syntactic contexts of all its non-stopwords, excluding dependencies with stopwords and words inside the term (see Figure 2.1). This technique allows us to represent in the same feature space both single terms and multiword expressions. In this experiment, we decided to exclude internal dependencies to minimize similarities between MWEs, which are similar syntactically, but not semantically, e. g. "proposition de loi", "proposition d'aide" and "proposition de travail". However, in some cases, it may be desirable to keep such internal dependencies.

The algorithm takes as input the syntactic dependencies SR, the index I with positions of the descriptors and the stoplists. At the first step, the algorithm creates an empty set of syntactic contexts B and a empty multiset C. An element of the multiset C is a tuple $\langle d, \beta \rangle$ which maps a term d and a syntactic context β. Then, the algorithm incrementally fills these two sets by checking each extracted syntactic tuple (lines 2-16). If the word w_i from the dependency $\langle w_i, p_i^{beg}, t, w_j, p_j^{beg} \rangle$ belongs to the term d, then we add the syntactic context $\langle t, w_i \rangle$ to the term d. Similarly, if the word w_j belongs to the term d, then we add the new syntactic context $\langle t, w_i \rangle$ to term d. The procedure will not add the syntactic context $\langle t, w_{context} \rangle$ to the term d if the context word $w_{context}$ is a part of term d or if it is a stopword (lines 12-13). The second part of the algorithm (lines 17-21) constructs the feature matrix $\mathbf{F}$ from the multiset C. Firstly, we set every element $f_{ij} \in \mathbf{F}$ of this matrix is equal to the number of times term d_i occurred in the context β_j (lines 18-19). Then, the line 20 normalizes the feature matrix with the Pointwise Mutual Information (Manning and Schütze, 1999, p.68). This is a commonly used normalization method for the distributional models (Heylen et al., 2008; Van de Cruys, 2010):

$$f'_{ij} = \log \frac{P(d_i, \beta_j)}{P(d_i)P(\beta_j)} \approx \log \frac{f_{ij}}{|d_i| \cdot |\beta_j|}. \tag{2.2}$$

In the formula, $|d_i|$ is the number of times term d_i occurred in the corpus and $|\beta_j|$ is the number of times the syntactic context β_j occurred in the corpus. After the normalization, every element of the feature matrix belongs to the interval between zero and one: $f'_{ij} \in [0; 1]$.

The procedure `GroupContexts` reduces sparsity of the distributional space by merging similar syntactic contexts, such as $\langle NMOD, 37\ millions\rangle$ and $\langle NMOD, 71\ millions\rangle$. The procedure merge features representing dates, sums of money, ordinal numbers, real numbers and percents. Finally, the procedure `RemoveContexts` deletes syntactic contexts occurred less than β^T times in the corpus: $B' = \{b_j \in B : |\beta_j| \geq \beta^T\}$. We present results of experiments with different values of this parameter in Section 2.2.4.

Algorithm 1: SDA-MWE measure: computation of the feature matrix $\mathbf{F}$.

Input: Dependencies $SR = \{SR_1, \ldots, SR_K\}$ of K documents; terms D; index I; stop part-of-speech SP; stopwords SW; dependency types T; threshold β^T.
Output: Distributional space B; Feature matrix $\mathbf{F}$.

$C \leftarrow \emptyset, B \leftarrow \emptyset, w_{context} \leftarrow$ "" $, w_{term} \leftarrow$ "" ;
// Calculating set of syntactic features B and multiset C
foreach *document* k *in corpus* **do**
  foreach $\left\langle w_i, p_i^{beg}, t, w_j, p_j^{beg}\right\rangle \in SR_k$ **do**
    if $\exists\left\langle k, d, p^{beg}, p^{end}\right\rangle \in I : p_i^{beg} \in [p^{beg}; p^{end}]$ **then**
      $w_{context} \leftarrow w_i$;
      $w_{term} \leftarrow d$;
    else if $\exists\left\langle k, d, p^{beg}, p^{end}\right\rangle \in I : p_j^{beg} \in [p^{beg}; p^{end}]$ **then**
      $w_{context} \leftarrow w_j$;
      $w_{term} \leftarrow d$;
    else
      continue ;
    if $w_{context} \notin w_{term}$ **and** $w_{term} \notin w_{context}$ **and** $t \in T$ **and**
    `GetPOS`$(w_{context}) \notin SP$ **and** $w_{context} \notin SW$ **then**
      $\beta \leftarrow \langle t_j, w_{context}\rangle$;
      $B \leftarrow B \cup \beta$;
      $C \leftarrow C \cup \langle w_{term}, \beta\rangle$;
// Calculating feature matrix $\mathbf{F}$
$\mathbf{F} \leftarrow \mathbf{0}_{|D|,|B|}$;
foreach $\langle d_i, \beta_j\rangle \in C$ **do**
  $f_{ij} \leftarrow f_{ij} + 1$;
`Normalize`$(\mathbf{F})$;
`GroupContexts`$(\mathbf{F}, B)$;
`RemoveContexts`$(\mathbf{F}, B, \beta^T)$;
return $B, \mathbf{F}$

Computation of Relations between Terms

We compute measures of semantic similarity between terms d_i and d_j with *Cosine Similarity* between their respective vectors (see Section 1.2.2):

$$sim(d_i, d_j) = s_{ij} = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{||\mathbf{f}_i||||\mathbf{f}_j||}. \tag{2.3}$$

We define a set of related terms to d as the set of its nearest neighbors. We extract relations between terms by thesholding the similarity matrix $\mathbf{S}$ with the threshold s^T (a kind of p-NN procedure described in Section 1.2.2): $\hat{R} = \{\langle d_i, d_j \rangle : s_{ij} \geq s^T\}$.

2.2.3 Evaluation

We assume that among several automatically-constructed thesauri $\{(C, \hat{R}_1), (C, \hat{R}_2), \ldots\}$, the best one is the most similar to the manually constructed thesaurus (C, R). In this experiment, we fix the vocabulary C and quantify the overlap between the extracted and original relations $|\hat{R}_i \cap R|$. In particular, we use the *exact precision* and *fuzzy precision* statistics. The exact precision is the number of automatically extracted relations found in the thesaurus, divided by the total number of extracted relations:

$$Precision^E = \frac{|\hat{R} \cap R|}{|\hat{R}|}. \tag{2.4}$$

The exact precision corresponds to the precision in the semantic relation extraction task (see Section 1.2.3). In this case, the golden standard is composed of the relations of the thesaurus R. The thesaurus is a handcrafted resource containing 3,986 relations between 2,514 concepts. This resource is not complete and subjective by definition. Let us illustrate the issue with the following example. In one of our experiments, the algorithm discovered that "foreign public act" is related to "private international law", "civil procedure" and "arbitration". Meanwhile, the thesaurus links "foreign public act" only to "legal act" and "foreign legislation". Thus, there is no overlap between these relations and the exact precision equals zero: $|\hat{R} \cap R| = \emptyset$.

To address this issue, we proposed the *fuzzy precision* measure. It takes into account both explicit and implicit relations of the thesaurus. The explicit relations are the original relations R. The implicit relations are the relations, which link terms in the thesaurus with a path of length $k \geq 2$. The technique is based on the observation that the thesaurus contains short paths between the descriptor "foreign public act" and the automatically discovered terms:

- foreign public act $\rightarrow$ foreign legislation $\rightarrow$ branch of law $\rightarrow$ private international law

- foreign public act → legal act → course of law → civil procedure
- foreign public act → legal act → course of law → civil procedure → arbitration

To calculate the fuzzy precision score we generate *fuzzy semantic relations* R^{Fk} and use them as a golden standard along with the original relations R. We generate fuzzy relations as follows:

1. Constructing adjacency matrix $\mathbf{W}$ of a set of hierarchical (R^{NT}) and associative (R^{RT}) relations of the thesaurus (C, R): $R = R^{NT} \cup R^{RT}$. An element of this matrix w_{ij} in three steps:
$$w_{ij} = \begin{cases} 2 & \text{if } \exists\langle d_j, d_i\rangle \in R^{NT} \\ 1 & \text{if } (\exists\langle d_i, d_j\rangle \in R^{NT}) \vee (\exists\langle d_i, d_j\rangle \in R^{RT}) \vee (\exists\langle d_j, d_i\rangle \in R^{RT}) \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$
2. Computing matrix of the shortest paths $\mathbf{P}$ between concepts of the thesaurus (C, R) with the Floyd's algorithm (Floyd, 1962). An element of this matrix p_{ij} contains length of the shortest path between concepts c_i and c_j.
3. Computing a set of fuzzy relations R^{Fk} between the terms. This set contains the pairs of terms connected in the thesaurus by a path with length less than or equal to k : $R^{Fk} = \{\langle c_i, c_j\rangle : p_{ij} \leq k\}$.

We constructed two fuzzy versions of the original thesaurus: R^{F3} and R^{F4}. The first set contained 80,641 pairs of concepts linked by a path in the thesaurus with length k less than or equal to 3. The second set contained 254,441 relations; it was constructed with the maximum path length of 4. The fuzzy precision measure is equal to the number of extracted relations found in the fuzzy thesaurus, divided by the total number of extracted relations:

$$Precision^{Fk} = \frac{|\hat{R} \cap R^{Fk}|}{|\hat{R}|}, \text{ where } k = \{3, 4\}. \quad (2.6)$$

2.2.4 Results

Table 2.2 presents relations between some descriptors extracted with the *SDA-MWE* method. The number in brackets is the length of the shortest path in the thesaurus (C, R) between descriptors from the first and the second columns.

We conducted several experiments varying the minimum frequency of syntactic context $\beta^T \in [0; 100]$ and the similarity matrix threshold $s^T \in [0; 1]$. Figure 2.2 shows that the best performance in terms of both exact and fuzzy precision is achieved with the similarity threshold s_T of 0.4. Furthermore, the threshold values s^T greater than 0.4 yield significantly

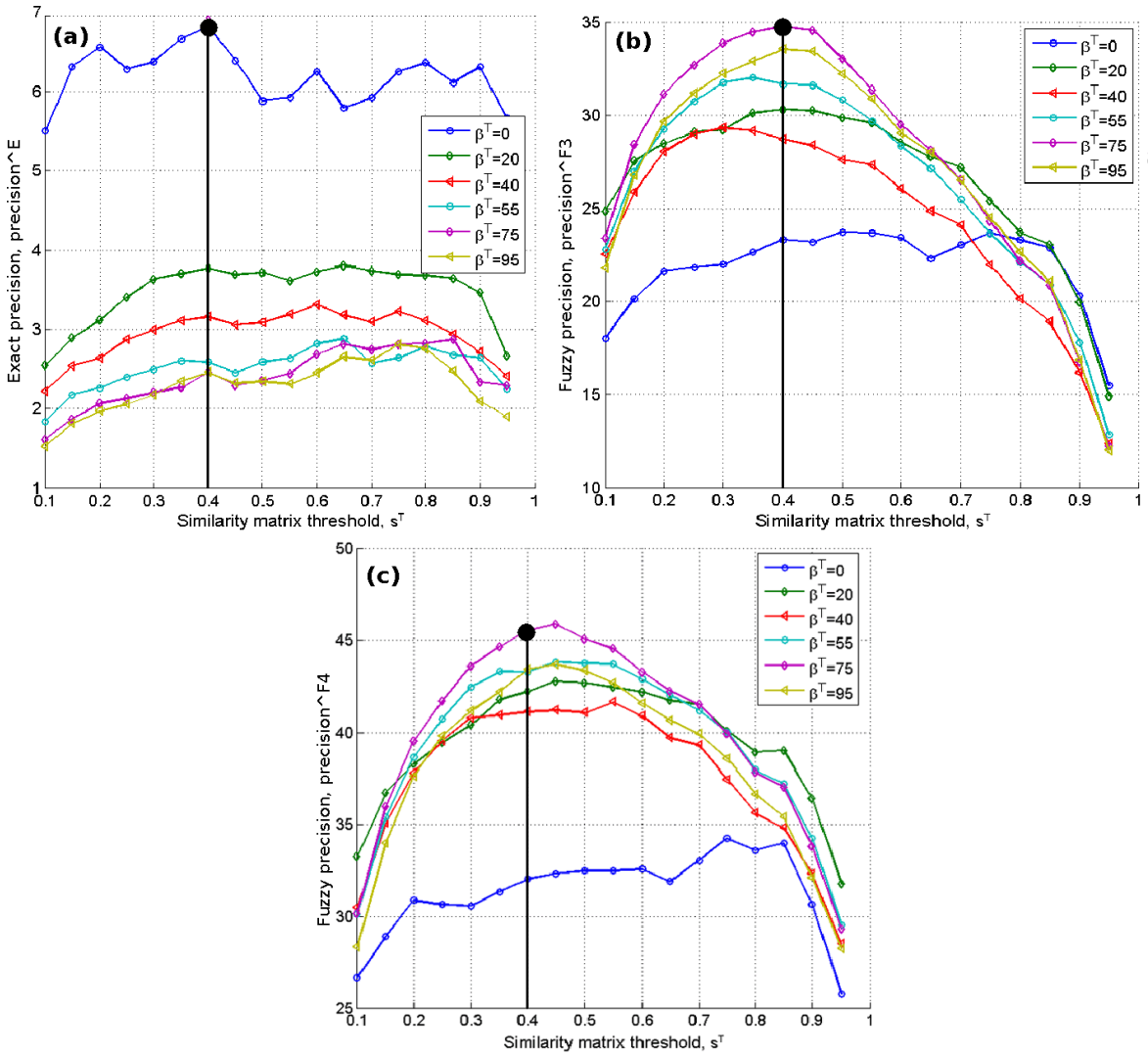


Figure 2.2: Performance of the semantic relation extraction method as evaluated with: (a) the exact precision statistic, (b) the fuzzy precision statistic for $k = 3$, (c) the fuzzy precision statistic for $k = 4$.

worse results. This is counter intuitive, as we would have expected a monotonically increasing curve. Such irregular behaviour is due to the sparsity of the syntactic representation – there are only few outlying scores greater than 0.4.

Figure 2.2 (a) shows that the extracted relations and the explicit thesaurus relations are completely different: $Precision^E \leq 0.07$. This result was obtained by the model using all the syntactic features: $\beta^T = 0$. This is likely to happen because of two reasons:

1. The distributional approach extracts co-hyponyms in the first place (see Chapter 3). On the other hand, terms in the thesaurus are linked with hyponyms (R^{NT}), synonyms (concepts) and associations (R^{RT}).

2. The corpus is small and we cannot be sure that the extraction is robust (compare Figure 3.7). Furthermore, our goal is to extract relations between the descriptors. Most of these terms are specific low-frequent multiword expressions. This makes the

task even more challenging.

The best results in terms of the fuzzy precision were obtained with the following parameters: $s_T = 0.4$ and $\beta^T = 75$. Figure 2.2 (b) shows that up to 35% of the extracted relations are linked in the original thesaurus with a path of lenght less or equal to three: $Precision^{F3} = 35\%$. On the other hand, Figure 2.2 (c) illustrates that up to 46% of the extracted relations are linked in the thesaurus with a path of lenght less or equal to four: $Precision^{F4} = 46\%$.

Term, d_i	**Related Term (Manual)**, d_j	**Related Term (Automatic)**, d'_j
administration of taxes	administration of the state	administration of the cadastre and the topography (2), state socio-educational center (8), public education (4), cultural institution (8), institute of hygiene and public health (7), state vineyard station (6)
admission to studies	school organization, education, admission to employment	archives of the state (9), certificate of teacher (6), program of studies (2)
medical assistance	medical organization	emergency medical services (1), medical analysis (6), medically assisted procreation (6), hygiene (6), wine institute (9), medical organization (1), medical profession (3), vaccination (5)
European election	election, political life, European parliament	legislative election (2)
unemployed person	unemployment, employment, employment administration	unemployment compensation (2)
education grants	school life, education	youth movement (11)
European community	European organisation, single European act, Yaounde agreement, Lome convention	European defense community (1), European atomic energy community (1), European coal and steel community (1), international economic partnership (2), country union (2)
school leaving certificate	diploma, promotion of students, school environment	foreign education certificate (2)
maternity leave	leave, number of hours, work	parental leave (3), work schedule (3)
South Africa	foreign country	Saudi Arabia (2), Bahamas (2), Belize (2), Colombia (2), Comoros (2), Congo (2), Djibouti (2), United Arab Emirates (2), Eritrea (2), federated states of Micronesia (2), Mexico (2), Gabon (2), guinea (2), equatorial guinea (2), Guyana (2), Kazakhstan (2)

Table 2.2: Comparison of automatically and manually constructed relations between terms of the thesaurus. We used the following parameters to generate these relations: $s^T = 0.4, \beta^T = 75$.

2.2.5 Summary

We proposed *SDA-MWE*, a method for semantic relation extraction based on the syntactic distributional analysis. The advantage of the method is that it can extract relations between multiword expressions (MWEs). In particular, it was used to extract relations between descriptors of a handcrafted thesaurus. However, the method has important limitations, in-

cluding low precision, the need to tune the threshold parameters and the fact that it does not return the type of the extracted relations.

This method cannot exactly reproduce relations from the original thesaurus, but can find terms linked with a short path in the original thesaurus. The experiments show a significant difference between the automatically and manually constructed relations. While many of the extracted relations are relevant, just a small fraction of them could be found in the original thesaurus (overlap of 7%, 35% or 46% depending on the type of golden standard). Nevertheless, our observations suggest that the proposed method can discover new relevant relations between the terms. We conclude that the method can be useful for automatic thesaurus construction, but its results may require a manual check.

2.3 DefVectors: A Similarity Measure Based on Definitions[5]

In this section we propose a new method which relies on the k-nearest neighbor procedures (see Section 1.2.2) and two semantic similarity measures based on definitions derived from the abstracts of Wikipedia [6] and Wiktionary [7]. In this section the method is tested on the semantic relation extraction task (see Section 1.2.3). Chapter 3 describes further experiments with the method on the standard benchmarks. We also present an open source system, which efficiently implements the technique.

A popular approach to relation extraction is based on the lexico-syntactic patterns (Hearst, 1992). The main drawbacks of this approach are the complexity of pattern construction and their language dependency. Methods based on distributional analysis (Lin, 1998b; Heylen et al., 2008) do not require any manual labor, but are less precise (Curran and Moens, 2002). Recently, the measures of semantic similarity based on Wikipedia have been proposed (Strube and Ponzetto, 2006; Gabrilovich and Markovitch, 2007; Zesch et al., 2008a). Wikipedia is attractive for text mining as it covers most of the topics in many languages. Furthermore, it is constantly updated by the users. These Wikipedia-based measures show excellent results.

The approach described in this section is an application of the Wikipedia-based measures to semantic relation extraction. The goal of the method proposed in this section is to discover a set of relations R between a set of input terms C (e. g., terms of a given domain). The main contributions of the work described in this section are two-fold:

- The new semantic relation extraction methods, which rely on the texts of Wikipedia articles, nearest neighbor procedures k-NN and mk-NN and similarity functions *Co-*

[5] The research presented in this section has been published as Panchenko et al. [5] and Панченко[13].
[6] Wikipedia, the free encyclopedia that anyone can edit: `http://www.wikipedia.org/`.
[7] Collaborative project for creating a free lexical database: `http://www.wiktionary.org/`.

sine Similarity and *Dice Coefficient* (see Section 1.2.2).

- An open source system, which efficiently implements the proposed methods. [8]

In Section 2.3.1, we introduce our approach to semantic relation extraction. First, we describe the data in Section 2.3.1. Next, we discuss the algorithms of semantic relation extraction (Section 2.3.1) and the measures of semantic similarity (Section 18). We present key details of the extraction system in Section 18. In Section 2.3.2, the experimental results are presented and discussed. Finally, Section 2.3.4 summarizes the experiments.

2.3.1 Method

Data and Preprocessing

The method requires a textual definition D for each input term $c \in C$. We use the data available from the `DBPedia.org` to gather such glosses of English words [9]. For each input term, a pair (c, d) is built, where term c is an exact title of a Wikipedia article and definition d is the abstract of this article. In the further experiments, presented in Section 3.4, we also deal with the definitions of Wiktionary. Wikipedia articles were preprocessed as follows. First, we removed all markup tags and special characters. Second, we performed lemmatization and part-of-speech tagging with the TreeTagger (Schmid, 1994). As a result, each word was represented as a triple `token#POS#lemma`, for instance `proved#VVN#prove`. An example of a definition in this format is provided below:

`axiom;` `in#IN#in traditional#JJ#traditional logic#NN#logic ,#,#, an#DT#an axiom#NN#axiom or#CC#or postulate#NN#postulate is#VBZ#be a#DT#a proposition#NN#proposition that#WDT#that is#VBZ#be not#RB#not proved#VVN#prove or#CC#or demonstrated#VVN#demonstrate but#CC#but considered#VVN#consider to#TO#to be#VB#be either#RB#either self-evident#JJ#self-evident ,#,#, or#CC#or subject#JJ#subject to#TO#to necessary#JJ#necessary decision#NN#decision .#SENT#.`

The experiments described in this section were conducted on a subset of Wikipedia articles. We prepared two datasets: [10]

- The first contains 775 words from the vocabulary of the BLESS dataset (see Section 1.2.3), which have a Wikipedia article with the corresponding title.
- The second contains 327,167 entries, corresponding to all single word titles of Wikipedia.

[8] `https://github.com/jgc128/defvectors`
[9] `http://downloads.dbpedia.org/3.7/en/long_abstracts_en.nt.bz2`
[10] `http://cental.fltr.ucl.ac.be/team/~panchenko/def/`

We limited scope of the study to the single terms, as it is possible to evaluate extractions between such words against various existing benchmarks (see Section 1.2.3). However, the method can be used to extract relations between multiword titles.

Senellart and Blondel (2008), Heylen et al. (2008) and some other researchers mention that methods based on the syntactic analysis, such as the one presented in Section 2.2, achieve higher results than methods based on part-of-speech tagging only. However, in our method we intentionally do not use the syntactic analysis for two reasons.

- Firstly, the computational complexity of the parsing algorithms is very high, making it difficult to process huge corpora.
- Secondly, such a complex linguistic analysis makes the method less robust. Prior research suggest that the quality of parsing in different languages is very different (Candito et al., 2010).
- Thirdly, the standard parsers make a lot of errors in the sentences that contain named entities and technical terms, the lexical units which are the most valuable in context of information extraction.

Algorithm of Semantic Relation Extraction

The semantic similarity measure *DefVectors* is based on the component analysis (Philippovich and Prokhorov, 2002; Kobozeva, 2009), which states that semantically similar words have similar definitions. The measure takes as an input a set of terms C, a dictionary of definitions D and some parameters such as the number of features β (see Algorithm 2). Additionally, the method takes as input a set of already known relations R.

The algorithm outputs a set of relations $\hat{R}$ between the input terms C and a $|C| \times |C|$ similarity matrix $\mathbf{S}$. Assume that the algorithm is processing the 5 following terms:

$$C = \{alligator, animal, building, house, telephone\}. \tag{2.7}$$

Its goal would be to recognize the two following relations

$$\hat{R} = \{\langle alligator, animal\rangle, \langle building, house\rangle\} \tag{2.8}$$

out of 10 possible pairs of terms:

$$\hat{R} = \{\langle alligator, animal\rangle, \langle alligator, building\rangle, \langle alligator, house\rangle, \ldots\}. \tag{2.9}$$

We proceed as follows to extract the relations and the similarities. First, we find a subset of definitions D_C corresponding to the input terms C (line 2). Next, the feature matrix $\mathbf{F}$ is constructed. Each term $c_i \in C$ is represented as a bag-of-words vector $\mathbf{f}_i$, derived from its definition (line 3). Optionally, the feature vectors can be normalized with Pointwise Mutual Information (lines 4-6).

Algorithm 2: Semantic similarity measure *DefVectors*.

Input: Terms C, Definitions D, Number of nearest neighbors k, Maximum number of features β, Known relations R, Similarity function sim, pmi – if true then PMI normalization else TF normalization, knn – if true then k-NN else mk-NN.

Output: Similarity matrix $\mathbf{S}$, Semantic relations $\hat{R}$.

// Constructing feature matrix ;
$D_C \leftarrow get\_definitions(C)$;
$\mathbf{F} \leftarrow construct\_fmatrix(D_C, \beta)$;
if pmi **then**
    $\mathbf{F} \leftarrow pmi(\mathbf{F})$;
// Calculating similarity matrix ;
for *i=1,|C|* **do**
    for *j=i,|C|* **do**
        $tmp \leftarrow sim(\mathbf{f}_i, \mathbf{f}_j)$;
        if $(\sum_{s_{ij} \neq 0 : j=1,|C|} 1 < k) \vee (tmp > \min_{j=1,|C|}(s_{ij}))$ **then**
            $s_{ij} \leftarrow tmp$
$\mathbf{S} \leftarrow update\_similarity(\mathbf{S}, R)$;
// Calculating relations ;
for *i=1,|C|* **do**
    forall the $j : s_{ij} \neq 0$ **do**
        if $knn \vee s_{ji} \neq 0$ **then**
            $\hat{R} \leftarrow \hat{R} \cup \langle c_i, c_j \rangle$
return $(\mathbf{S}, \hat{R})$;

Next, we compute pairwise similarities between all the input terms (lines 8-15) with the similarity function sim (line 10). We rely either on the *Dice Coefficient* or *Cosine Similarity* between vectors of definitions (see Section 1.2.2). We keep the number of non-zero elements in each row of $\mathbf{S}$ equals k, which lets us minimize the memory footprint. Finally, the pairwise similarities between terms can be corrected with the function $update\_similarity$ if R is not void (line 16). This routine assigns the highest scores to the known relations [11]:

$$s'_{ij} = \begin{cases} 1 & \text{if } \langle c_i, c_j \rangle \in R \\ s_{ij} & \text{otherwise} \end{cases} \tag{2.10}$$

The goal of the last step is to obtain relations $\hat{R}$ from the similarity matrix $\mathbf{S}$ with the k-NN or the mk-NN procedures. We rely on the nearest neighbors procedures as this is the extraction paradigm chosen in this work (see Section 1.2.2). The thresholding on the similarity (p-NN) proven to be very sensitive to the threshold value (see Section 2.2). That is why in this experiment, we use the k-NN and mk-NN techniques. The k-NN simply prints the k nearest neighbor terms of each term. In contrast, mk-NN establishes a relation $\langle c_i, c_j \rangle$ only if the words are mutual neighbors (lines 18-24). Thus, mk-NN filters out those relations extracted by k-NN which are not mutually related. The use of mk-NN is justified by the assumption that the semantic relations are symmetric (see Section 1.1.1).

Measures of Semantic Similarity

Function `similarity` (line 6) in the algorithms k-NN and mk-NN calculates a pairwise similarity of two terms $c_i, c_j \in C$ from their definitions $d_i, d_j \in D$. The larger the value of semantic similarity, the closer the "sense" of the terms. Two similarity functions are considered here. The first is the *Dice Coefficient* of the definitions d_i, d_j of the terms c_i, c_j (Section 1.2.2):

$$sim(c_i, c_j) = \frac{2 \cdot |d_i \cap d_j|}{|d_i| + |d_j|}. \tag{2.11}$$

Here the numerator is the number of the common words in the definitions; $|d_j|$ is the number of unique words in the definition d_j. The second measure is the cosine between vectors $\mathbf{f}_i, \mathbf{f}_j$ of definitions d_i, d_j representing terms c_i, c_j:

$$sim(c_i, c_j) = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{||\mathbf{f}_i|| \cdot ||\mathbf{f}_j||} = \frac{\sum_{k=1,N} f_{ik} f_{jk}}{\sqrt{\sum_{k=1,N} f_{ik}^2} \sqrt{\sum_{k=1,N} f_{jk}^2}}. \tag{2.12}$$

[11] In this section we do not use this option, but it is employed in the experiments reported in Chapter 3.

Here f_{ik} is the frequency of the lemma c_k in the definition d_i. Both similarity measures use lemmas (e. g., `animals#NNS#animal`) and use does not the stopwords found in the definitions. For both similarity measures, the terms are represented with nouns (NN, NNS, NP), verbs (VV, VVN, VVP) and adjectives (JJ) of their definitions. All other words are omitted.

Semantic Relation Extraction System

The system is a console application implemented in C++ and available for Windows and Linux platforms (32/64 bits). The main functions of the program are:

- loading files of stopwords and input terms C;
- loading the file with definitions D taking into account the stopwords;
- computation of the pairwise semantic similarities between the input terms C;
- building the list of the semantic relations R.

In order to achieve high performance, we map each word to a numerical identifier. This procedure significantly reduces running time of the program. The system extensively uses the Standard Template Library [12] and the Boost Library. [13]

2.3.2 Results

We investigated the algorithms k-NN and mk-NN with the two measures described above and with various numbers of nearest neighbors k (see Figure 2.3). As one may expect, the number of extracted relations linearly depends on the number of nearest neighbors k both for k-NN and mk-NN. The number of extracted relations depends little on the similarity measure type. The key difference between the two measures is that *Cosine Similarity* takes into account frequencies, while *Dice Coefficient* does not. The little difference in the results is likely to be due to the fact that the definitions are short. Thus, frequency information does not contribute a lot to the result.

The algorithm k-NN extracts more relations than the mk-NN for the same value of k. It happens because the mk-NN filters out pairs of terms which are not mutual nearest neighbors. According to our experiments, for a vocabulary size $|C|$ of 775 words, mk-NN filters around 50-70% of the relations extracted by k-NN. Generally, the number of filtered relations will depend on the number of the terms $|C|$ and the value of k.

[12] `http://www.cplusplus.com/reference/stl/`
[13] `http://www.boost.org/`.

We estimated the precision of the extraction between 775 terms for both algorithms with $k = 2$. In order to measure the precision, we manually labeled the files with the extracted relations. The precision was computed as the number of correctly extracted relations to the number of extracted relations (see Section 1.2.3). The results are presented in Table 2.4. The examples of extracted relations between a set of 775 words with the mk-NN procedure ($k = 2$) and the *Dice Coefficient* are presented in Table 2.3 [14]:

Term, c_i	**Term**, c_j
acacia	pine
aircraft	rocket
alcohol	carbohydrate
alligator	coconut
altar	sacristy
object	library
object	pattern
office	crew
onion	garlic
saxophone	violin
saxophone	clarinet
tongue	mouth
watercraft	boat
watermelon	berry
weapon	warship
wolf	coyote
wood	paper

Table 2.3: Examples of extracted relations between a set of 775 terms with the mk-NN procedure ($k = 2$) and the Dice Coefficient.

Due to the big number of extracted relations (see Figure 2.3), it is hard to estimate extraction precision for the big values of k. We expect the precision to decrease for values of $k > 2$. We recommend to use the number of the nearest neighbors $k \in [1; 10]$. Evaluation of the *DefVectors* measure on the standard benchmarks (Sections 1.2.3) is presented in Section 3.6.

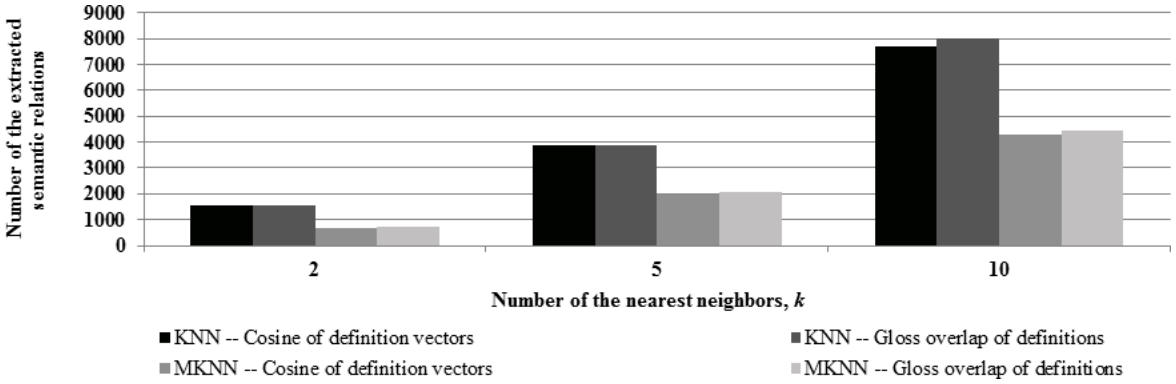


Figure 2.3: Dependence of the number of extracted relations on the number of nearest neighbors k.

[14] The full list of the extracted relations with this configuration is available at `http://cental.fltr.ucl.ac.be/team/~panchenko/def/results-775/overlap_mknn_2.csv`

Algorithm	Similarity Measure	# Extracted Relations	# Correct Relations	Precision
k-NN	Cosine Similarity	1548	1167	0.754
k-NN	Dice Coefficient	1546	1176	0.754
mk-NN	Cosine Similarity	652	499	0.763
mk-NN	Dice Coefficient	724	603	**0.833**

Table 2.4: Precision of relation extraction for 775 terms with the k-NN and mk-NN ($k = 2$).

2.3.3 Discussion

In this section, we discuss systems similar to *DefVectors*. The automatic thesaurus construction system SEXTANT (Grefenstette, 1994), based on the distributional analysis, extracts relations between words with precision of around 75%. However, one of the most similar systems to ours is WikiRelate! (Strube and Ponzetto, 2006). It achieves a correlation with human judgments of 0.59. The main differences between the *DefVectors* and this system are the following:

- The source code of WikiRelate! is not available, while the binary version is available only for research purposes. The source code of *DefVectors* is open source.
- *DefVectors* can compute similarity not only between texts of Wikipedia, but also between any other definitions encoded in the corresponding format.
- *DefVectors* uses either on the *Cosine Similarity* or on the *Dice Coefficient*, while WikiRelate! relies on the *Dice Coefficient* normalized with the hyperbolic tangent function.
- *DefVectors* does not use the category lattice of Wikipedia.
- *DefVectors* uses known semantic relations to update similarity scores.

Zesch et al. (2008a) proposed another measure of semantic similarity based on Wikipedia. The *DefVectors* is similar to this technique, but differs from it in three aspects:

- In *DefVectors*, each term is represented as a bag-of-words vector, while the measures of Zesch et al. (2008b) represent terms as concept vectors.
- *DefVectors* uses known semantic relations to update similarity scores.
- *DefVectors* can compute similarity not only between texts of Wikipedia, but also between any other definitions represented in the corresponding format.

Gabrilovich and Markovitch (2007) proposed yet another original Wikipedia-based measure called Explicit Semantic Analysis (*ESA*). The key feature of this approach is that it represents a text in the space of all Wikipedia articles. Nakayama et al. (2007) suggested to infer associations between words from the hyperlink structure of Wikipedia articles. Finally, Milne et al. (2006) proposed a way to extract synonyms, hypernyms and associations from Wikipedia category lattice and other structural and navigational elements of Wikipedia.

2.3.4 Summary

We proposed a method for semantic relation extraction from texts of Wikipedia with k-NN and mk-NN procedures and two semantic similarity measures. The preliminary experiments showed that the best results (precision of 83% on the set of 775 terms) are obtained with the method based on the mk-NN procedures and the *Dice Coefficient*. We also presented an open source system, which efficiently implements the proposed technique.

The method is able to compute relations between a huge number of terms, each of which is represented by a title of a Wikipedia article. Thus, it can potentially be used to extract relations between 3.8 million terms in English Wikipedia and 17 million terms in other 282 languages of Wikipedia. The only language-dependent resources used in the method are the stoplist, the part-of-speech tagger and the lemmatizer. These resources are available for most European languages for free. Finally, *DefVectors* can extract relations from other sources of definitions, such as traditional dictionaries or Wiktionary, if these data are provided in the corresponding format.

2.4 PatternSim: A Similarity Measure Based on Lexico-Syntactic Patterns[15]

This section presents a novel semantic similarity measure based on lexico-syntactic patterns, such as those proposed by Hearst (1992). We evaluate its correlation with human judgements, as well as its performance on the semantic relation ranking and semantic relation extraction tasks (see Section 1.2.3).

As mentioned in Section 2.1, three well-established approaches to semantic similarity are based respectively on semantic networks, dictionaries and corpora. Network- and dictionary-based measures achieve high precision, but suffer from limited coverage as they rely on manually-crafted semantic resources. On the other hand, corpus-based measures provide decent recall, but suffer from lower precision as most of them rely on a simple representation based on the Vector Space Model. To overcome coverage issues of the resource-based techniques while maintaining their precision, we adapt an approach to semantic similarity, based on lexico-syntactic patterns. Bollegala et al. (2007) proposed to compute semantic similarity with automatically harvested *lexical* patterns. In our approach, we rather rely on explicitly specified *lexico-syntactic* patterns, such as those proposed by Hearst (1992).

Contributions of this section are two-fold:

- First, we present a novel corpus-based semantic similarity measure *PatternSim* based

[15]The research presented in this section has been published as Panchenko et al [3].

on lexico-syntactic patterns. The measure performs comparably to the baseline measures, but requires no semantic resources such as WordNet or dictionaries.

- Second, we describe an open source implementation of the proposed approach, that has been made available to the community.

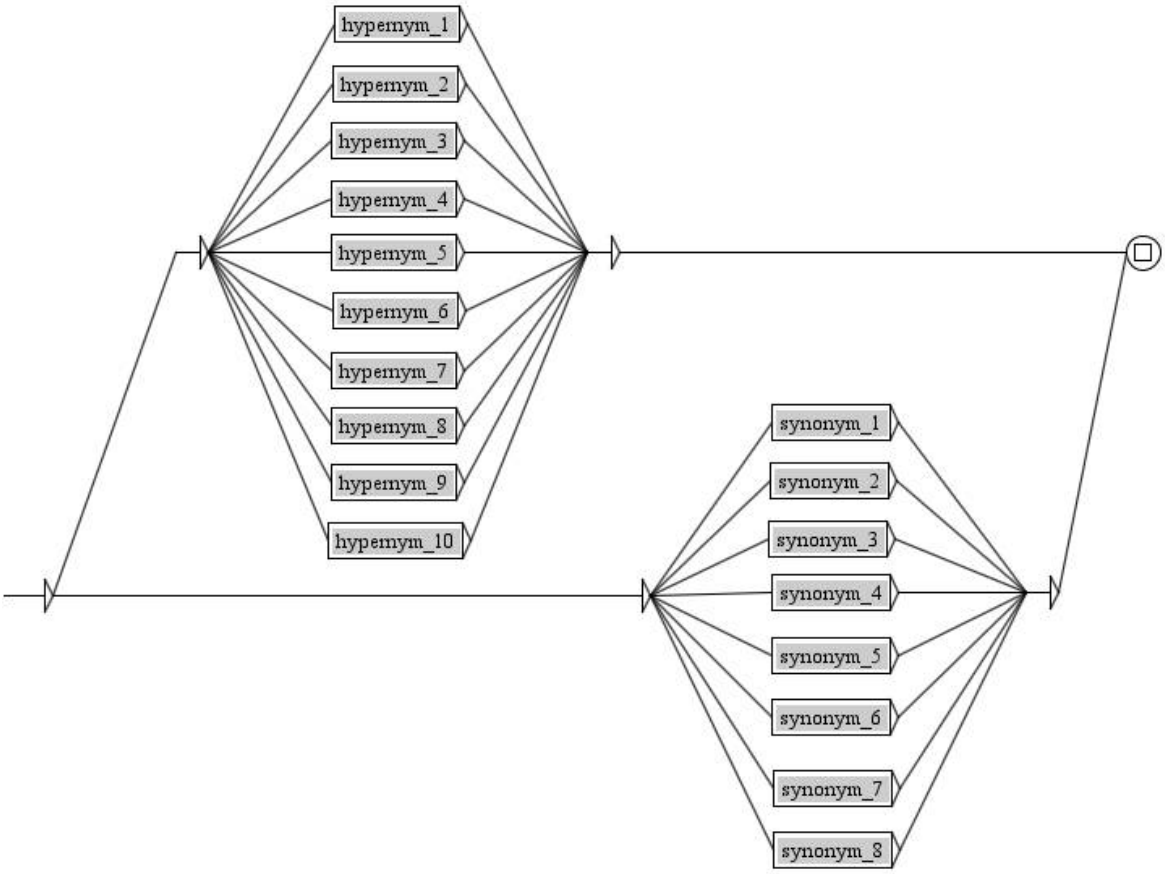


Figure 2.4: The main Finite State Transducer (a "graph"), which combines the 18 lexico-syntactic patterns. This graph is a cascade of the subgraphs, each encoding one pattern.

2.4.1 Lexico-Syntactic Patterns

A lexico-syntactic pattern relies on lexical information and on syntactic categories. We extended the set of the 6 classical Hearst (1992) patterns (1-6) with 12 further patterns (7-18), which aim at extracting hypernymic and synonymic relations:

```
1. such {NP=hyper} as {NP=hypo}, {NP=hypo}[,] and/or {NP=hypo};
2. {NP=hyper} such as {NP=hypo}, {NP=hypo}[,] and/or {NP=hypo};
3. {NP=hypo}, {NP=hypo}[,] or other {NP=hyper};
4. {NP=hypo}, {NP=hypo}[,] and other {NP=hyper};
5. {NP=hyper}, including {NP=hypo}, {NP=hypo}[,] and/or {NP=hypo};
6. {NP=hyper}, especially {NP=hypo}, {NP=hypo} [,] and/or {NP=hypo};
7. {NP=hyper}: {NP=hypo}, [{NP=hypo},] and/or {NP=hypo};
8. {NP=hypo} is DET ADJ.Superl {NP=hyper};
```

```
9. {NP=hyper}, e. g., {NP=hypo}, {NP=hypo}[,] and/or {NP=hypo};
10. {NP=hyper}, for example, {NP=hypo}, {NP=hypo}[,] and/or {NP=hypo};
11. {NP=syn}, i. e.[,] {NP=syn};
12. {NP=syn} (or {NP=syn});
13. {NP=syn} means the same as {NP=syn};
14. {NP=syn}, in other words[,] {NP=syn};
15. {NP=syn}, also known as {NP=syn};
16. {NP=syn}, also called {NP=syn};
17. {NP=syn} alias {NP=syn};
18. {NP=syn} aka {NP=syn}.
```

This scheme is only able to capture similarities between noun phrases (NPs). This is not an issue, as in this work we focus on synonyms, hypernyms and co-hyponyms (see Section 1.1.1). The patterns are encoded in finite-state transducers (FSTs) with the help of the corpus processing tool *Unitex* (Paumier, 2003) [16]. Figure 2.4 depicts the main FST, which combines all the patterns in one automaton. Figure 2.5 illustrates FSTs of three patterns. Here each box in gray color denotes a sub-FST.

Patterns are based on linguistic knowledge and thus provide a more precise representation than co-occurrences or bag-of-word models. *Unitex* makes it possible to build negative and positive contexts, to exclude meaningless adjectives and so on. We listed above the key features of the patterns. However, they are more complex as they take into account variation of natural language expressions. Thus, FST-based patterns can achieve higher recall than string-based patterns such as those used by Bollegala et al. (2007).

2.4.2 Semantic Similarity Measures

The outline of the similarity measure *PatternSim* is provided in Algorithm 3 [17]. The method takes as input a set of terms of interest C and a corpus D. Semantic similarities between these terms are returned in a $C \times C$ sparse similarity matrix $\mathbf{S}$. An element of this matrix s_{ij} is a real number within the interval $[0; 1]$ which represents the strength of semantic similarity.

Name	# Documents	# Tokens	# Lemmas	Size
WaCky	2,694,815	$2{,}026 \cdot 10^9$	3,368,147	5.88 Gb
ukWaC	2,694,643	$0.889 \cdot 10^9$	5,469,313	11.76 Gb
WaCky + ukWaC	5,387,431	$2.915 \cdot 10^9$	7,585,989	17.64 Gb

Table 2.5: Corpora used by the PatternSim measure.

As a first step, lexico-syntactic patterns are applied to the input corpus D (line 1). In our

[16] http://igm.univ-mlv.fr/~unitex/

[17] The method described above is implemented in an open source system *PatternSim* available under the conditions of LGPLv3: https://github.com/cental/patternsim.

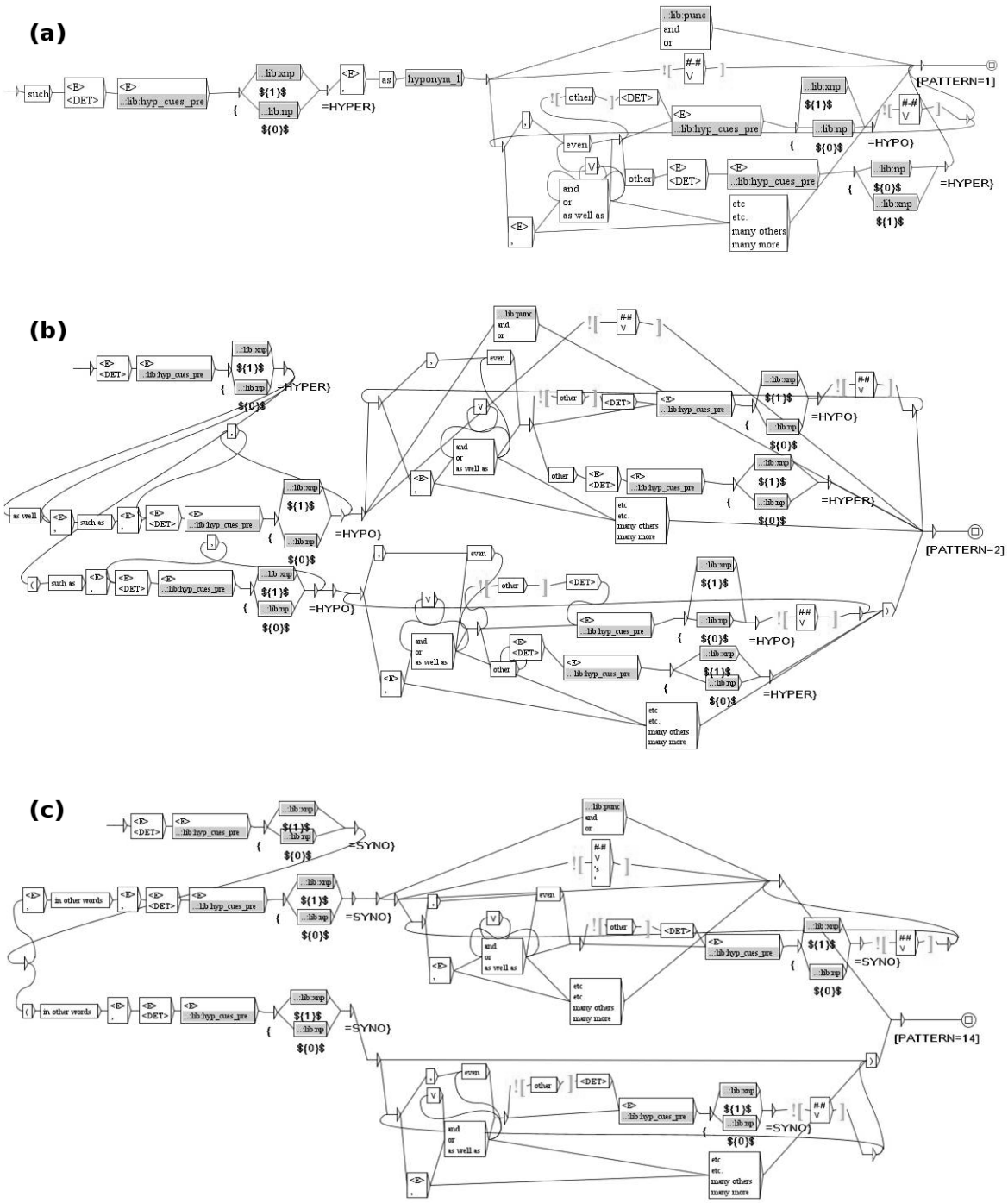


Figure 2.5: Examples of the Unitex graphs: (a) hypernymy/co-hyponymy extraction with the pattern #1; (b) hypernymy/co-hyponymy extraction with the pattern #2; (c) synonymy extraction with the pattern #14. Here subgraphs are marked with gray. <E> defines zero; <DET> defines determiners; symbols and letters outside of the boxes are markup tags.

experiments we used three corpora presented in Table 2.5: *WaCky*, *ukWaC* and the combination of both (Baroni et al., 2009). *WaCky* is a dependency-parsed corpus of English Wikipedia abstracts. *ukWaC* is a dependency-parsed a corpus of Web pages. However, the *PatternSim*, uses nothing, but the surface form of these texts.

Applying a cascade of FSTs to a corpus is a memory and CPU consuming operation. To make processing of these huge corpora feasible, we split the entire corpus into blocks of 250 Mb. Processing such a block took around one hour on an Intel i5 M520@2.40GHz with 4 Gb of RAM. This is the most computationally heavy operation of Algorithm 3. The method retrieves all the concordances matching the 18 patterns. Each concordance is marked up in a specific way:

```
such diverse {[occupations]=hyper} as {[doctors]=hypo}, {[engineers]=↩
    ↩hypo} and {[scientists]=hypo}[PATTERN=1]
such {non-alcoholic [sodas]=hyper} as {[root beer]=hypo} and {[cream ↩
    ↩soda]=hypo}[PATTERN=1]
{traditional[food]=hyper}, such as {[sandwich]=hypo}, {[burger]=hypo}, ↩
    ↩and {[fry]=hypo}[PATTERN=2]
{[mango]=hypo},{[ pineapple]=hypo}, {[jackfruit]=hypo} and other{[fruit↩
    ↩]=hyper}[PATTERN=4]
{primitive [snake]=hyper}, such as {[boa]=hypo} and {[python]=hypo}[↩
    ↩PATTERN=2]
{[France]=hypo},{[ Belgium]=hypo} and other{European [country]=hyper}[↩
    ↩PATTERN=4]
such{[big city]=hyper} as{[Kiev]=hypo},{[Moscow]=hypo}, [Leningrad]=↩
    ↩hypo},{[Kharkov]=hypo}.[PATTERN=1]
{[fruit]=hyper} such as{[apple]=hypo},{[blackberry]=hypo}, {[blueberry↩
    ↩]=hypo}, {[cantaloupe]=hypo}, {[cherry]=hypo}, {[cranberry]=hypo},↩
    ↩ {[grape]=hypo}, {[pear]=hypo}, {[plum]=hypo}, {[raspberry]=hypo},↩
    ↩ and{[strawberry]=hypo}[PATTERN=2]
```

Curly brackets mark the noun phrases, which are in the semantic relation; nouns and compound nouns stand between square brackets. We extracted 1,196,468 concordances K of this type from *WaCky* corpus and 2,227,025 concordances from ukWaC – 3,423,493 in total. For the next step (line 2), the nouns in the square brackets are lemmatized with the DELA dictionary[18], which consists of around 300,000 simple and 130,000 compound words. The concordances which contain at least two terms from the input vocabulary C are selected (line 3). Subsequently, the similarity matrix $\mathbf{S}$ is filled with frequencies of pairwise extractions (line 4). At this stage, we fill the matrix with the number of co-occurrences of terms in the square brackets within the same concordance e_{ij}. Finally, the word pairs are re-ranked with one of the methods described below (line 5). Once the reranking is done, the similarity

[18] Available at `http://infolingu.univ-mlv.fr/`

scores are mapped to the interval $[0; 1]$ as follows (line 6):

$$\acute{\mathbf{S}} = \frac{\mathbf{S} - min(\mathbf{S})}{max(\mathbf{S}) - min(\mathbf{S})}. \tag{2.13}$$

Algorithm 3: Similarity measure *PatternSim*.

Input: Terms C, Corpus D
Output: Similarity matrix, $\mathbf{S}$ $[C \times C]$

1 $K \leftarrow extract\_concord(D)$;
2 $K_{lem} \leftarrow lemmatize\_concord(K)$;
3 $K_C \leftarrow filter\_concord(K_{lem}, C)$;
4 $\mathbf{S} \leftarrow get\_extraction\_freq(C, K)$;
5 $\mathbf{S} \leftarrow rerank(\mathbf{S}, C, D)$;
6 $\mathbf{S} \leftarrow normalize(\mathbf{S})$;
7 **return** $\mathbf{S}$;

Efreq

In this case, no re-ranking is done. Thus, the semantic similarity s_{ij} between c_i and c_j is equal to the frequency of extractions e_{ij} between the terms $c_i, c_j \in C$ in a set of concordances K.

Efreq-Rfreq

This re-ranking formula penalizes terms that are strongly related to many words. In this case, semantic similarity of terms equals:

$$s_{ij} = \frac{2 \cdot \mu_e \cdot e_{ij}}{e_{i*} + e_{*j}}, \tag{2.14}$$

where $e_{i*} = \sum_{j=1}^{|C|} e_{ij}$ is a number of concordances containing word c_i. Similarly $e_{*j} = \sum_{i=1}^{|C|} e_{ij}$ is the number of concordances including word c_j. The μ_e is an expected number of semantically related words per term ($\mu_e = 20$). Essentially, μ_e is introduced to improve readability of the similarity scores:

$$s_{ij} \gg 1, \quad \text{terms } c_i \text{ and } c_j \text{ are highly related} \tag{2.15}$$

$$s_{ij} \ll 1, \quad \text{terms } c_i \text{ and } c_j \text{ are highly unrelated} \tag{2.16}$$

Efreq-Rnum

This formula also reduces the weight of terms which have many relations to other words. Here we rely on the number of extractions b_{i*} with a frequency superior to β: $b_{i*} = \sum_{j:e_{ij}\geq\beta} 1$ and $b_{*j} = \sum_{i:e_{ij}\geq\beta} 1$. Semantic ranking is calculated in this case as follows:

$$s_{ij} = \frac{2 \cdot \mu_b \cdot e_{ij}}{b_{i*} + b_{*j}}, \tag{2.17}$$

where $\mu_b = \frac{1}{|C|} \sum_{i=1}^{|C|} b_{i*}$ is an average number of related words per term. Similarly, to the previous formula, μ_b is used to improve readability of the similarity scores. We experiment with values of $\beta \in \{1, 2, 5, 10\}$.

Efreq-Cfreq

This formula penalizes relations to general words, such as "item". According to this formula, similarity equals:

$$s_{ij} = \frac{P(c_i, c_j)}{P(c_i)P(c_j)}, \tag{2.18}$$

where $P(c_i, c_j) = \frac{e_{ij}}{\sum_{ij} e_{ij}}$ is the extraction probability of the pair $\langle c_i, c_j \rangle$, $P(c_i) = \frac{f_i}{\sum_i f_i}$ is the probability of the word c_i and f_i is the frequency of c_i in the corpus. We use the original corpus D and the corpus of concordances K to derive f_i.

Efreq-Rnum-Cfreq

The goal of this formula is to normalize the score at the same time with number of extracted relations (as in the *Efreq-Rnum*) and the term frequencies (as in the *Efreq-Cfreq*):

$$s_{ij} = \frac{2 \cdot \mu_b}{b_{i*} + b_{*j}} \cdot \frac{P(c_i, c_j)}{P(c_i)P(c_j)}. \tag{2.19}$$

Efreq-Rnum-Cfreq-Pnum

This formula integrates extra information to the previous one about the number of patterns $p_{ij} = \overline{1, 18}$ extracted the given pair of terms $\langle c_i, c_j \rangle$. Some patterns, such as #5 and #7, are especiall prone to errors. The relations extracted independently by several patterns tend to be more precise than those extracted only by a single pattern. The p_{ij} variable follows the "robustness via redundancy" principle. While one pattern may produce a big number of false positive extractions, it is unlikely that each of 18 patterns will extract many false positives for the given pair of terms $\langle c_i, c_j \rangle$. We use the root to smooth the effect of this

variable for the large values of p_{ij}. Therefore, the similarity of terms equals:

$$s_{ij} = \sqrt{p_{ij}} \cdot \frac{2 \cdot \mu_b}{b_{i*} + b_{*j}} \cdot \frac{P(c_i, c_j)}{P(c_i)P(c_j)}. \tag{2.20}$$

2.4.3 Evaluation and Results

We evaluated the similarity measure proposed above on three tasks – correlations with human judgments about semantic similarity, ranking of word pairs and extraction of semantic relations (see Section 1.2.3) [19].

Correlation with Human Judgments

The first evaluation is based on the MC, RG and WordSim datasets (Section 1.2.3). The quality of a measure is assessed with Spearman's correlation between vectors of scores (see Table 2.6). The first part of the table reports on scores of 12 baseline similarity measures. We compare the proposed technique with the WordNet-based measures *WuPalmer, LecockChodorow* and *Resnik* (see Sections 2.1 and 3.2); the corpus-based measures *BDA, SDA* and *LSA* (see Sections 2.1, 3.3.1 and 3.3.3), the definition-based measures *DefVectors, GlossVectors* and *ExtendedLesk* (see Sections 2.1, 3.4) and three measures based on the *WikiRelate* (Strube and Ponzetto, 2006).

The second part of the table presents various modifications of our measure based on lexico-syntactic patterns. The first two modifications are based on *WaCky* and *ukWaC* corpora, respectively (see Table 2.5). All the remaining *PatternSim* measures use both corpora (*WaCky+ukWaC*) as, according to our experiments, they provide better results. Correlations of the *PatternSim* measures are comparable to those of the baselines. In particular, the proposed measures outperform most of the baselines on the *WordSim* dataset achieving a correlation of 0.520.

While *PatternSim* performs similarly to the measures based on WordNet and dictionary glosses, it requires no handcrafted semantic resources. Development of the extraction patterns needs some times and skills. However, we argue that the effort required to handcraft such patterns is significantly lower than the effort required to build a resource like WordNet. In our case, about 4-6 man-weeks was spent to develop and debug the extraction grammars. Furthermore, once constructed, the patterns can be used to harvest relations from several corpora.

[19] Results of the evaluation: `http://cental.fltr.ucl.ac.be/team/panchenko/sim-eval`

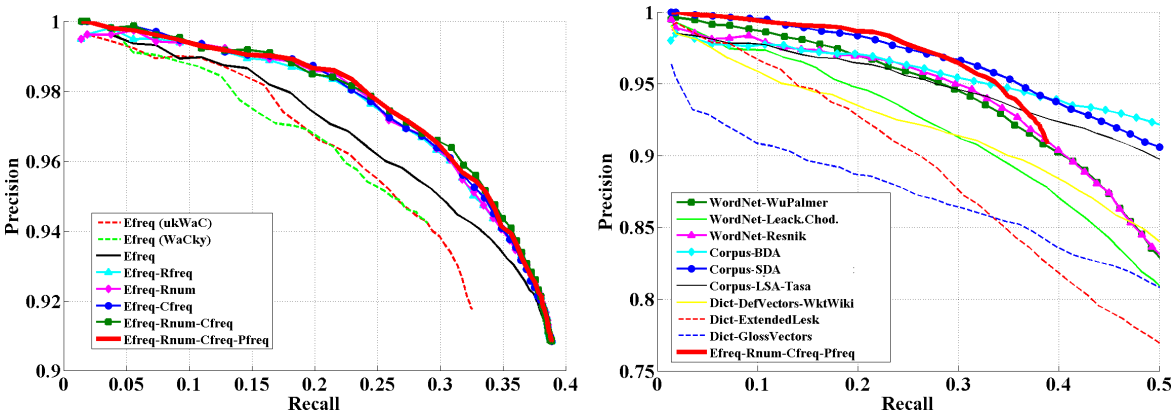


Figure 2.6: Precision-Recall graphs calculated on the BLESS (hypo,cohypo,mero,attri,event) dataset: (a) variations of the PatternSim measure; (b) the best PatternSim measure as compared to the baseline similarity measures.

Semantic Relation Ranking

In this section we consider the semantic relation ranking task (see Section 1.2.3). Table 2.6 and Figure 2.6 present the performance of baseline and pattern-based measures on the BLESS and SN datasets. Precision of the similarity scores learnt from the *WaCky* corpus is higher than that obtained from the *ukWaC*, but the recall of *ukWaC* is better since this corpus is larger (see Figure 2.6 (a)). Thus, in accordance to the previous evaluation, the biggest corpus *WaCky+ukWaC* provides better results than the *WaCky* or the *ukWaC* alone. Ranking relations with extraction frequencies (*Efreq*) provides results that are significantly worse than any re-ranking strategies. On the other hand, the difference between various re-ranking formulae is small with a slight advantage for *Efreq-Rnum-Cfreq-Pnum*.

The performance of the *PatternSim (Efreq-Rnum-Cfreq-Pnum)* measure is comparable to those of the baselines (see Figure 2.6 (b)). Furthermore, in terms of precision, it outperforms the 9 baselines, including syntactic distributional analysis (*SDA*). However, its recall is significantly lower than the baselines because of the sparsity of the pattern-based approach. Indeed, the similarity of terms can only be computed if they co-occur in the corpus within an extraction pattern. For instance, the *PatternSim* curve stops at the precision-recall point $(0.925, 0.4)$ as the measure cannot provide non-zero similarity scores to all pairs from the BLESS dataset. However, one may continue the curve by drawing a line from this point to the point $(1.0, 0.0)$.

On the other hand, *PatternSim* achieves both high recall and precision on the BLESS benchmark containing only hypernyms and co-hyponyms (see Table 2.6). Therefore, the proposed measure extracts mostly these relations. This is an expected result as our patterns were designed to extraction of synonyms, hyponyms and co-hyponyms.

Similarity Measure	MC	RG	WS	BLESS (hypo,cohypo,mero,attri,event)				SN (syn, hypo, cohypo)				BLESS (hypo, cohypo)			
	ρ	*ρ*	*ρ*	*P(10)*	*P (20)*	*P(50)*	*R(50)*	*P(10)*	*P(20)*	*P(50)*	*R(50)*	*P(10)*	*P(20)*	*P(50)*	*R(50)*
Random	*0.056*	*-0.047*	*-0.122*	0.546	0.542	0.544	0.522	0.504	0.502	0.499	0.498	0.271	0.279	0.286	0.502
WordNet-WuPalmer	0.742	0.775	0.331	0.974	0.929	0.702	0.674	0.982	0.959	0.766	**0.763**	0.977	0.932	0.547	**0.968**
WordNet-Leack.Chod.	0.724	**0.789**	0.295	0.953	0.901	0.702	0.648	**0.984**	0.953	0.757	0.755	0.951	0.897	0.542	0.957
WordNet-Resnik	0.784	0.757	0.331	0.970	0.933	0.700	0.647	0.948	0.908	0.724	0.722	**0.968**	0.938	0.542	0.956
Corpus-BDA	0.693	0.782	0.466	0.971	0.947	0.836	**0.772**	0.974	0.932	0.742	0.740	0.908	0.828	0.502	0.886
Corpus-SDA	**0.790**	0.786	0.491	0.985	0.953	0.811	0.749	0.978	0.945	0.751	0.743	0.979	0.921	0.536	0.947
Corpus-LSA-Tasa	0.694	0.605	0.566	0.968	0.937	0.802	0.740	0.903	0.846	0.641	0.609	0.877	0.775	0.467	0.824
Dict-DefVectors-WktWiki	0.759	0.754	0.521	0.943	0.905	0.750	0.679	0.922	0.887	0.725	0.656	0.837	0.769	0.518	0.739
Dict-GlossVectors	0.653	0.738	0.322	0.894	0.860	0.742	0.686	0.932	0.899	0.722	0.709	0.777	0.702	0.449	0.793
Dict-ExtenedLesk	0.792	0.718	0.409	0.937	0.866	0.711	0.657	0.952	0.873	0.655	0.654	0.873	0.751	0.464	0.820
WikiRelate-Gloss	0.460	0.460	0.200	–	–	–	–	–	–	–	–	–	–	–	–
WikiRelate-Leack.Chod.	0.410	0.500	0.480	–	–	–	–	–	–	–	–	–	–	–	–
WikiRelate-SVM	–	–	**0.590**	–	–	–	–	–	–	–	–	–	–	–	–
Efreq (WaCky)	0.522	0.574	0.405	0.971	0.950	0.942	0.289	0.930	0.912	0.897	0.306	0.976	0.937	0.923	0.626
Efreq (ukWaC)	0.384	0.562	0.411	0.974	0.944	0.918	0.325	0.922	0.905	0.869	0.329	0.971	0.926	0.884	0.653
Efreq	0.486	0.632	0.429	0.980	0.945	0.909	**0.389**	0.938	0.915	0.866	**0.400**	0.976	0.929	0.865	**0.739**
Efreq-Rfreq	**0.666**	**0.739**	0.508	0.987	0.955	0.909	**0.389**	0.951	0.922	0.867	**0.400**	0.983	0.940	0.865	**0.739**
Efreq-Rnum	0.647	0.720	0.499	0.989	0.955	0.909	**0.389**	0.951	0.922	0.867	**0.400**	0.983	0.940	0.865	**0.739**
Efreq-Cfreq	0.600	0.709	0.493	0.989	0.956	0.909	**0.389**	0.949	0.920	0.867	**0.400**	0.986	0.948	0.865	**0.739**
Efreq-Cfreq (concord.)	**0.666**	**0.739**	0.508	0.986	0.954	0.909	**0.389**	0.952	0.921	0.867	**0.400**	0.984	0.944	0.865	**0.739**
Efreq-Rnum-Cfreq	0.647	0.737	0.513	0.988	0.959	0.909	**0.389**	**0.953**	0.924	0.867	**0.400**	**0.987**	0.947	0.865	**0.739**
Efreq-Rnum-Cfreq-Pnum	0.647	0.737	**0.520**	**0.989**	0.957	0.909	**0.389**	0.952	0.924	0.867	**0.400**	0.985	0.947	0.865	**0.739**

Table 2.6: Performance of the baseline similarity measures as compared to various modifications of the PatternSim measure on human judgments datasets (MC, RG, WS) and semantic relation datasets (BLESS and SN). All correlations with human judgements (MC, RG and WordSim) are significant.

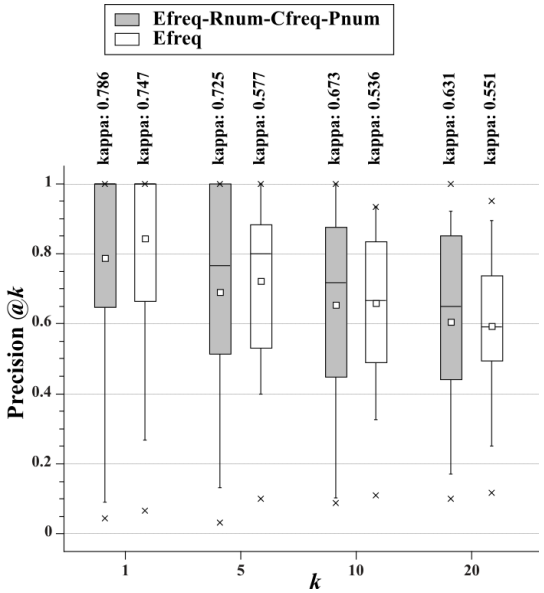


Figure 2.7: Semantic relation extraction: precision at k.

Semantic Relation Extraction

We evaluated relations extracted with the *Efreq* and the *Efreq-Rnum-Cfreq-Pnum* measures for 49 words (vocabulary of the RG dataset). Three annotators indicated whether the terms were semantically related or not. We calculated for each of the 49 words the extraction precision for $k = \{1, 5, 10, 20\}$. Figure 2.7 shows the results of this evaluation. For the *Efreq* measure, average precision indicated by white boxes varies between 0.792 (the top relation) and 0.594 (the 20 top relations), whereas it goes from 0.736 (the top relation) to 0.599 (the 20 top relations) for the *Efreq-Rnum-Cfreq-Pnum* measure. The inter-raters agreement between the annotators in terms of Fleiss's kappa (Fleiss, 1971) was substantial (0.61-0.80) or moderate (0.41-0.60). Thus, the proposed measure performs well not only on

the specific benchmarks, such as the correlations with human judgements, but also are able to extract semantic relations from text.

2.4.4 Summary

In this section, we presented a similarity measure based on manually-crafted lexico-syntactic patterns, which achieves a correlation with human judgments up to 0.739. The measure was evaluated on five ground truth datasets (MC, RG, WordSim353, BLESS, SN) and on the task of semantic relation extraction. Our results have shown that the measure provides results comparable to the baseline WordNet-, dictionary-, and corpus-based measures and does not require semantic resources. We argue that the effort required to craft the lexico-syntactic patterns is much less as compared to the work needed to manually construct a resource. The proposed ranking formulae could be refined with a supervised model proposed in Chapter 4. A supervised model could help to select parameter values (α and β) and to combine different factors ($e_{ij}, e_{i*}, P(c_i), P(c_i, c_j), p_{ij}$, etc.) in one model.

2.5 Conclusion

In this chapter, we provided a state-of-the-art on the single semantic similarity measures. Next, we presented three novel single similarity measures based respectively on the syntactic distributional analysis (Section 2.2), definitions of Wiktionary and Wikipedia (Section 2.3) and lexico-syntactic patterns (Section 2.4).

Experimental results described in this chapter suggest that each measure has its pros and cons. The distributional measures have a good coverage, but their precision is not always meet the needs of the applications, such as the automatic thesaurus construction. For a limited vocabulary, measures based on definitions are more precise. However, coverage of such techniques is limited by the size of the dictionary. The measures based on lexico-syntactic patterns have high precision. Their coverage is superior to the dictionary-based techniques, but still significantly lower than that of distributional measures. We conclude that measures based on different resources may be potentially complementary. This idea is further developed in the next chapter.

Chapter 3

Comparison of Network-, Corpus-, and Definition-Based Similarity Measures[1]

I often say that when you can measure what you are speaking about and express it in numbers you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind: it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of science, whatever the matter may be.

– William Thomson, 1891

In this chapter, we evaluate a wide range of single semantic similarity measures on the tasks of correlation with human judgments and semantic relation ranking (see Section 1.2.3).

The existing single similarity measures differ both in the the kinds of information they use and in the ways this information is transformed into a similarity score (see Section 2.1). They rely either on semantic networks (Resnik, 1995), text corpora (Lin, 1998b), Web as a corpus (Cilibrasi and Vitanyi, 2007), dictionaries (Lesk, 1986) or encyclopedia (Zesch et al., 2008a). Prior research suggests that measures based on these sources of information are complementary (Sahlgren, 2006; Heylen et al., 2008; Panchenko, 2011). In this chapter, further investigated the intuition that the baselines are highly complementary.

The contributions of this chapter are two-fold:

1. We present a comparative study of heterogeneous baseline measures. Several authors already compared existing approaches, but we perform a study on a large scale, as we compare 37 similarity measures based on corpora, definitions and networks. In

[1]The research presented in this chapter was published as Panchenko [7], Panchenko [9], Panchenko [10].

particular, we are the first to compare these measures on the semantic relation ranking task (see Section 1.2.3).

2. We go further than most of the surveys, such as Lee (1999), Agirre et al. (2009) or Ferret (2010) and compare the measures with respect to the semantic relation types they provide (hypernyms, meronyms, etc.). We report empirical relation distributions for each measure and propose a way to find the measures with the most and the least similar relation distributions.

3.1 Related Work

This section deals with the existing surveys of semantic similarity measures. Refer to Section 2.1 for a state-of-the-art of the single similarity measures. Senellart and Blondel (2008) present an overview of the research on semantic similarity.

The distributional semantic similarity measures are relatively well-studied. Lee (1999) and Ferret (2010) conducted comprehensive comparisons of such techniques. Curran and Moens (2002) evaluated 9 distributional measures and 14 weight functions. Authors suggest that a combination of the *Jaccard Index* (see Section 1.2.2) with the t-test weighting yields the best results on the semantic relation extraction task (see Section 1.2.3): a $Precision@5$ of 0.52 and a $Precision@10$ of 0.45. Van de Cruys (2010) evaluated syntactic (*SDA*) and bag-of-words (*BDA*) distributional measures against the Dutch WordNet. This study suggests that the syntactic context models (*SDA*) are the best for the extraction of tight synonym-like similarity. Heylen et al. (2008) compared general performances and relation distributions of the distributional measures. Sahlgren (2006) evaluated syntagmatic and paradigmatic bag-of-word models (*BDA*).

We know less about comparative performances of the measures based on different resources. Budiu et al. (2007) compared techniques based on the text corpora (*LSA* and *GLSA*(Matveeva, 2007)) and Web as a corpus (*PMIIR*). The authors found that *GLSA* performs better on the synonymy tests, while *PMIIR* works better on the human judgment datasets. Agirre et al. (2009) compared 3 WordNet-based and 20 distributional measures.

Some studies compared the measures in the context of NLP applications. For instance, Mihalcea et al. (2006) applied to the text similarity task measures based on the Web as a corpus (*PMIIR*), text corpora (*LSA*) and a semantic network. The authors found that *PMIIR* and *Resnik* are the best corpus- and network-based measures correspondingly. Budanitsky and Hirst (2006) reported that the *JiangConrath* measure is the best network-based measure for the task of spelling correction. Patwardhan and Pedersen (2006) evaluated six network-based measures on the task of word sense disambiguation and reported the same finding. Syntactic distributional analysis was used by Grefenstette (1994) to induce a thesaurus from

a text corpus.

In the following sections, we present the semantic similarity measures used in this comparative study. These measures rely on semantic networks, text corpora, Web as a corpus or dictionary definitions. According to the best of our knowledge, no study benchmarks all these techniques on several task and datasets.

3.2 Network-Based Measures

Network-based measures use a semantic network in order to calculate similarities. Some of the measures also use counts from a corpus. In the literature, network-based measures are often referred as knowledge-based measures (Mihalcea et al., 2006). We tested six such measures based on *WordNet* (Miller, 1995b) and *SemCor* corpus (Miller et al., 1993) (Pedersen et al., 2004): *InvEdgeCount* (Jurafsky and Martin, 2009, p. 687), *Leacock-Chodorow* (Leacock and Chodorow, 1998), *Resnik* (Resnik, 1995), *JiangConrath* (Jiang and Conrath, 1997), *Lin* (Lin, 1998a) and *WuPalmer* (Wu and Palmer, 1994) [2].

These measures use the following variables to compute the similarities:

- $len(c_i, c_j)$ – length of the shortest path in the network between terms c_i and c_j.
- $len(c_i, c_{ij})$ – length of the shortest path from c_i to the lowest common subsumer (LCS) of c_i and c_j, denoted as c_{ij}. A LCS of two terms c_i and c_j is a term c_{ij} on the shortest path between c_i and c_j that is closest to the root of the network c_{root}:
$$\nexists c'_{ij} : (len(c_i, c_j) < \infty) \wedge (len(c'_{ij}, c_{root}) < len(c_{ij}, c_{root})). \quad (3.1)$$
- $P(c)$ – the probability of term c estimated from a corpus.

Let us illustrate the notion of the lowest common subsumers on the semantic network depicted in Figure 3.1. Lengths of the shortest paths between the nodes "car" and "food" and the nodes "beef" and "pork" are equivalent. The LCS of the nodes "car" and "food" is the node "object", while the LCS of the nodes "beef" and "pork" is the node "meat". Therefore, semantic similarity of the terms "beef" and "pork" is greater as the depth of the node "meat" is greater than the depth of the node "object". This is so, as the leaves of a semantic network represent the most specific concepts, while its root represent the most general concept.

The *InvEdgeCount* measure relies on the length of the shortest path between the terms c_i and c_j to calculate their similarity s_{ij}:

$$s_{ij} = len(c_i, c_j)^{-1}. \quad (3.2)$$

[2] We used the WordNet::Similarity tool: `http://wn-similarity.sourceforge.net/`.

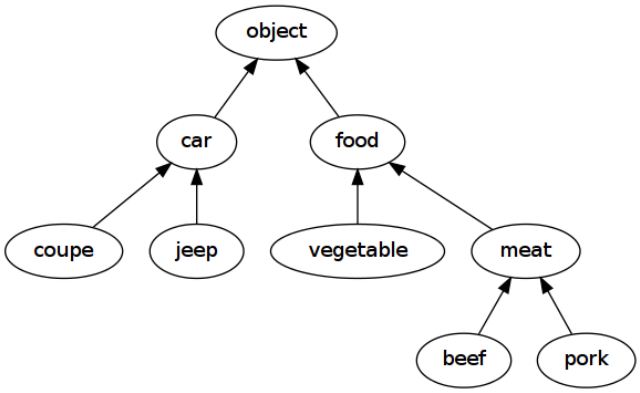


Figure 3.1: Lowest common subsumers in a semantic network.

The *LeacockChodorow* measure is very similar to the *InvEdgeCount*:

$$s_{ij} = -\log \frac{len(c_i, c_j)}{2h}. \tag{3.3}$$

Here h is the height of the network. The *Resnik* measure relies on the probability of the LCS of two terms:

$$s_{ij} = -\log P(c_{ij}). \tag{3.4}$$

The *JiangConrath* distance relies on the probabilities of terms and their LCS:

$$d_{ij} = 2\log P(c_{ij}) - (\log P(c_i) + \log P(c_j)). \tag{3.5}$$

The *Lin* measure uses the same information as the *JiangConrath*, but transforms it into a similarity score in a different way:

$$s_{ij} = \frac{2\log(P(c_{ij}))}{\log(P(c_i) + \log(P(c_j))}. \tag{3.6}$$

The *WuPalmer* measure relies on lengths of the shortest path between terms, their lowest common subsumer and the root term:

$$s_{ij} = \frac{2len(c_r, c_{ij})}{len(c_i, c_{ij}) + len(c_j, c_{ij}) + 2 \cdot len(c_r, c_{ij})}. \tag{3.7}$$

The complexity of the network-based measures is mainly bounded by the computation time of the shortest paths between the nodes of the network: $len(c_i, c_j)$. The baseline algorithm (Dijkstra, 1959) finds a shortest path in $O(|E| + |C| \cdot log|C|)$ time, where $|E|$ is the number of relations and $|C|$ is the number of terms in the semantic network.

A limitation of the network-based measures is that coverage of these measures is limited by the coverage of the semantic network. In our case, similarities can only be calculated

between the 155,287 English terms encoded in the WordNet 3.0. For instance, since the named entity "ACL" is not present in WordNet, no relations between "ACL" and other words can be retrieved.

3.3 Corpus-Based Measures

We experimented with 19 measures that calculate similarity of terms based on statistics derived from a corpus. The first 3 of them are based on the *PatternSim* measures introduced in Section 2.4:

1. *PatternSim-Efreq*,
2. *PatternSim-EfreqCfreq*,
3. *PatternSim-EfreqCfreqRnumPnum*.

They extract similarity scores from text with a set of lexico-syntactic patterns. The other 16 measures are based on distributional analysis, Web as a corpus approach or Latent Semantic Analysis.

3.3.1 Distributional Measures

These 13 corpus-based measures rely on the distributional analysis (Sahlgren, 2006; Curran, 2003) [3]. Algorithm 4 presents a pseudocode for the distributional measures. First, a distributional measure builds a feature matrix $\mathbf{F}$ from a corpus D, such that each term $c_i \in C$ is represented with a row-vector $\mathbf{f}_i$ (line 1). We experiment with two types of distributional measures:

- The measures which rely on the Bag-of-words Distributional Analysis (*BDA*) construct the feature matrix $\mathbf{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_n)^T$ with the context window technique (Van de Cruys, 2010). In this case, a term is represented with a bag of lemmas from a context window, passing a stop-word filter (around 900 words) and a part-of-speech filter (nouns, adjectives and verbs are kept).
- The measures which rely on Syntactic Distributional Analysis (*SDA*) construct the feature matrix $\mathbf{F}$ with the syntactic context technique (Lin, 1998b; Van de Cruys, 2010). This measure is essentially equivalent to the technique discussed in Section 2.2. The only differences are that the measure used in this section works with English and does not support multiword expressions. Let the term c_i = "cat" be linked with syntactic dependency dt_j = OBJ with the word w_k = "catch". Syntactic context of the term c_i is a bag of dependency-word pairs linked to it $\{\langle dt_j, w_k\rangle : w_k \notin Stoplist \wedge dt_k \in DT\}$, where DT is a set of dependency types used by the measure.

Algorithm 4: Distributional similarity measures *BDA*/*SDA*.

Input: Vocabulary C, Corpus D, Number of features β, Min.term frequency θ, if *BDA* then k is the size of context window, else if *SDA* then k is the number of syntactic dependencies.

Output: Similarity matrix, $\mathbf{S}$ $[C \times C]$

1 $\mathbf{F} \leftarrow construct\_feature\_matrix(C, D, \beta, \theta, k)$;
2 $\mathbf{F} \leftarrow pmi(\mathbf{F})$;
3 $\mathbf{S} \leftarrow sim(\mathbf{F})$;
4 **return** $\mathbf{S}$;

An element of the feature matrix $f_{ij} \in \mathbf{F}$ is equal to the number of times term c_i was represented with the feature f_j. The feature matrix $\mathbf{F}$ is normalized with Pointwise Mutual Information (line 2):

$$f_{ij} = log \frac{P(c_i, f_j)}{P(c_i)P(f_j)} \approx log \frac{f_{ij}}{\sum_j f_{ij} \sum_i f_{ij}}. \tag{3.8}$$

Finally, the similarity between the terms c_i and c_j is computed as similarity of their feature vectors $\mathbf{f}_i, \mathbf{f}_j$ (line 3). The similarity is calculated with one of the following distances (see Section 1.2.2): Cosine Similarity, Jaccard Index, Manhattan distance and Euclidean distance. In particular, we tested the following 13 distributional similarity measures:

- *BDA-sent-Cos, BDA-sent-Jaccard, BDA-sent-Manhattan* and *BDA-sent-Euclidean* are based on Bag-of-words Distributional Analysis, context window size of one sentence, 10,000 most frequent features and four similarity functions mentioned above;
- *BDA-1-Cos, BDA-2-Cos, BDA-3-Cos, BDA-5-Cos, BDA-8-Cos, BDA-10-Cos* are based on Bag-of-words Distributional Analysis, a symmetric context window of different sizes (1, 2, 3, 5, 8 and 10 words, respectively), 5,000 most frequent features and the Cosine Similarity.
- *SDA-6-Cos, SDA-9-Cos, SDA-21-Cos* are based on Syntactic Distributional Analysis, 100,000 most frequent syntactic features and the Cosine Similarity. Here *SDA-6-Cos* relies on 6 types of syntactic dependencies [4]: $DT_6 = \{$ NMOD, SBJ, OBJ, COORD, AMOD, IOBJ$\}$; *SDA-9-Cos* relies on 9 types of syntactic dependencies: $DT_9 = \{$ NMOD, ADV, SBJ, OBJ, VMOD, COORD, AMOD, PRN, IOBJ $\}$; and *SDA-21-Cos* relies on 21 types of syntactic dependencies: $DT_{21} = \{$ NMOD, P, PMOD, ADV, SBJ, OBJ, VMOD, COORD, CC, VC, DEP, PRD, AMOD, PRN, PRT, LGS, IOBJ, EXP, CLF, GAP $\}$.

[3] In these experiments, we used an in-house implementation of the distributional measures.
[4] See `http://www.maltparser.org/` for details.

In our experiments, we use two general English corpora (Baroni et al., 2009): *WaCky* (800M tokens) and *PukWaC* (2,000M tokens). These corpora are tagged with *TreeTagger* (Schmid, 1994) and dependency-parsed with *MaltParser* (Hall et al., 2011).

3.3.2 Web-Based Measures

Web-based measures use the Web as a corpus in order to calculate similarities. They rely on the number of times terms co-occur in documents indexed by a Web search engine. In particular, web-based measures rely on the number of documents (hits) h_i returned by the system for the query "c_i", the number of hits h_{ij} returned by the query "c_i AND c_j" and the number of documents indexed by the system M. We use two web-based measures: Normalized Google Distance (*NGD*) introduced by Cilibrasi and Vitanyi (2007):

$$s_{ij} = \frac{max(log(h_i), log(h_j)) - log(h_{ij})}{log(M) - min(log(h_i), log(h_j))}, \tag{3.9}$$

and Pointwise Mutual Information Information Retrieval (*PMIIR*) proposed by Turney (2001):

$$s_{ij} = \log \frac{P(c_i, c_j)}{P(c_i)P(c_j)} = \log \frac{\frac{h_{ij}}{\sum_{i,j} h_{ij}}}{\frac{h_i}{\sum_{i,j} h_{ij}} \frac{h_j}{\sum_{i,j} h_{ij}}} \approx \log \frac{h_{ij}}{h_i h_j}. \tag{3.10}$$

We tested 9 web-based measures [5]:

- *NGD-Bing, NGD-Yahoo, NGD-YahooBoss, NGD-Google* and *NGD-GoogleWiki* are based on Normalized Google Distance formula and respectively on *Bing*, *Yahoo*, *YahooBoss*, *Google*. Additionally we test *Google* over the domain `wikipedia.org`.
- *PMIIR-Bing, PMIIR-YahooBoss, PMIIR-Google* and *PMIIR-GoogleWiki* are based on PMIIR formula and respectively on *Bing*, *YahooBoss*, *Google* or *Google* over the domain `wikipedia.org`.

In addition to these 9 web-based measures, we test 2 corpus-based measures available via the *MSR* service:

- *NGD-Factiva* is a measure based on the Factiva corpus [6] and *NGD* formula, mentioned above (Veksler et al., 2008).
- *PMIIR-Factiva* is a measure based on the Factiva corpus and the *PMIIR* formula.

The complexity of the web-based measures is mainly bounded by the maximum number of queries. As of July 2012, *Bing* allows up to 5,000 queries per month for free and asks 2$ for

[5] Our own system was used in the experiments with measures based on *Bing* (`http://www.bing.com/toolbox/bingdeveloper/`) and *YahooBoss* (`http://developer.yahoo.com/search/boss/`). Measures of Semantic Relatedness (MSR) web service (`http://cwl-projects.cogsci.rpi.edu/msr/`) was used for the measures based on *Google* and *Yahoo!*.

[6] `https://global.factiva.com`

1,000 queries; *Google* allows 100 queries per day for free or 1,000 queries for 5$; *Yahoo* asks 0.80$ for 1,000 queries. This economical issue limits the use of such techniques. In our experiments, we calculated only the similarity scores between the pairs from the evaluation datasets. Thus, to use web-based measures a free access to a search engine is desirable. One of the key advantages of such measures is that they cover a huge of vocabulary in multiple of languages.

3.3.3 Latent Semantic Analysis

The *LSA-Tasa* measure relies on the Latent Semantic Analysis applied on the TASA corpus (Veksler et al., 2008). These are the standard *LSA* settings used in the original paper (Landauer and Dumais, 1997) and available via the *MSR* service. The main steps of the Latent Semantic Analysis are the following:

1. Representing the corpus D as an $N \times M$ term-document matrix $\mathbf{F}$, where each column-vector $\mathbf{f}$ represents a document in an M-dimensional vector space. An element $f_{ij} \in \mathbf{F}$ of this matrix contains frequency of the word w_i in the document $d_j \in D$.

2. Normalization of the matrix $\mathbf{F}$ with TF-IDF (Aizawa, 2003):
$$f'_{ij} = \frac{f_{ij}}{\sum_i f_{ij}} \cdot log \frac{|D|}{|d \in D : w_i \in d|}, \tag{3.11}$$
where f_{ij} is a frequency of the word w_i in document d_j, $|D|$ is the number of documents and $|d \in D : w_i \in d|$ is the number of documents where the term w_i appears.

3. Singular value decomposition of the matrix $\mathbf{D}$ [7]:
$$\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{3.12}$$
where $\mathbf{U}$ is an $M \times M$ matrix which columns are the orthogonal eigenvectors of $\mathbf{D}\mathbf{D}^T$, $\mathbf{V}^T$ is an $N \times N$ matrix which columns are the orthogonal eigenvectors of $\mathbf{D}^T\mathbf{D}$ and $\mathbf{\Sigma}$ is an $M \times N$ diagonal matrix:
$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{nn} \end{pmatrix}. \tag{3.13}$$
The i-th element on the diagonal $\sigma_{ii} = \sqrt{\lambda_i}$, where λ_i is an eigenvalue of $\mathbf{D}\mathbf{D}^T$. The eigenvalues are ordered, such that $\lambda_i \geq \lambda_{i+1}$. Figure 3.2 provides an example of a singular value decomposition of a term-document matrix $\mathbf{D}$. Here each row-vector of matrix $\mathbf{U}$ represents a term.

4. Low-rank approximation of the matrix $\mathbf{U}$ with a reduced $M \times k$ matrix $\mathbf{U}_k$ by retaining

[7] Description of this step is adapted from (Manning et al., 2008, p.374)

only the first k column of the $\mathbf{U}$. If $k = 2$, then we will obtain the following matrix:

$$\mathbf{U}_k = \begin{pmatrix} -0.7071 & 0.0000 \\ -0.5000 & -0.7071 \\ -0.5000 & 0.7071 \end{pmatrix}. \tag{3.14}$$

5. Calculation of similarities between terms c_i and c_j as a cosine between respective columns of $\mathbf{U}_k$ denoted as $\mathbf{u}_i^k$ and $\mathbf{u}_i^k$:

$$s_{ij} = \frac{\mathbf{u}_i^k \cdot \mathbf{u}_j^k}{||\mathbf{u}_i^k||||\mathbf{u}_j^k||}. \tag{3.15}$$

For the example listed above, we will obtain the following similarity matrix:

$$\mathbf{S} = \begin{pmatrix} 1.0000 & 0.5774 & 0.5774 \\ 0.5774 & 1.0000 & -0.3333 \\ 0.5774 & -0.3333 & 1.0000 \end{pmatrix}. \tag{3.16}$$

In the experiments described in this chapter, we used the implementation of LSA from the mentioned above MSR service [8].

$$\text{If } \mathbf{D} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}, \text{ then } \mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \text{ where}$$

$$\mathrm{U} = \begin{pmatrix} -0.7071 & 0.0000 & 0.7071 \\ -0.5000 & -0.7071 & -0.5000 \\ -0.5000 & 0.7071 & -0.5000 \end{pmatrix}, \Sigma = \begin{pmatrix} 1.8478 & 0 & 0 & 0 & 0 \\ 0 & 1.4142 & 0 & 0 & 0 \\ 0 & 0 & 0.7654 & 0 & 0 \end{pmatrix},$$

$$\mathbf{V} = \begin{pmatrix} -0.6533 & -0.5000 & 0.2706 & 0 & -0.5000 \\ -0.2706 & 0.5000 & -0.6533 & 0 & -0.5000 \\ -0.6533 & 0.5000 & 0.2706 & 0 & 0.5000 \\ 0 & 0 & 0 & 1.0000 & 0 \\ -0.2706 & -0.5000 & -0.6533 & 0 & 0.5000 \end{pmatrix}.$$

Figure 3.2: An example of singular vector decomposition of an $M \times N$ term-document matrix D.

3.4 Definition-Based Measures

We experimented with six measures which rely on explicit definitions of terms. The first four are variations of the *DefVectors* measure presented in Section 2.3:

1. *DefVectors-Wkt-1000* relies on 1,000 most frequent bag-of-word features extracted

[8] An alternative implementation of the LSA service is available at `http://lsa.colorado.edu/`

from definitions of Wiktionary [9]. This measure relies also on the lexico-semantic relations explicitly encoded in the Wiktionary (synonyms, hypernyms, etc.). The parameters of the *DefVectors* used in this experiment are as follows (see Algorithm 2): $\beta = 1000$, $D = Wiktionary$, $sim = cos$, $pmi = true$, $R = Wiktionary$, $knn = true$.

2. *DefVectors-Wkt-2500* relies on 2,500 most frequent features extracted from the Wiktionary: $\beta = 2500, D = Wiktionary$, $sim = cos$, $pmi = true$, $R = Wiktionary$, $knn = true$.

3. *DefVectors-WktWiki-1000* relies on 1,000 most frequent features extracted from the Wiktionary and the Wikipedia: $\beta = 1000, D = Wiktionary + Wikipedia$, $sim = cos$, $pmi = true$, $R = Wiktionary$, $knn = true$.

4. *DefVectors-WktWiki-2500* relies on 2,500 most frequent features extracted from the Wiktionary and the Wikipedia: $\beta = 2500, D = Wiktionary + Wikipedia$, $sim = cos$, $pmi = true$, $R = Wiktionary$, $knn = true$.

We built the dictionary of Wiktionary definitions D as follows. A definition of each term c was composed of glosses, examples, quotations, related words and categories found in the Wiktionary. [10] We merged glosses which correspond to different senses. The syntax- and etymology-related categories such as "English nouns" or "Japanese proper names" were removed with a stoplist of 94 words, such as "noun" or "esperanto". The dictionary of Wikipedia definitions D was built in the same way as suggested in Section 2.3. Each definition of the term c was composed of the abstract of the Wikipedia article with the corresponding title.

The next two measures rely on the definitions and relations from WordNet [11]. The key difference between Wiktionary- and WordNet-based measures is that the latter uses definitions of related terms. The *Extended Lesk* (Banerjee and Pedersen, 2003) measure relies on the gloss similarity of terms c_i and c_j as well as gloss similarity of all terms related to c_i and c_j:

$$s_{ij} = \sum_{c_i \in C_i} \sum_{c_j \in C_j} sim_g(c_i, c_j), \tag{3.17}$$

where sim_g is a gloss-based similarity measure and set C_i includes concept c_i and all concepts directly related to it. The sim_g is defined by the authors as follows:

> "When comparing two glosses, we define an overlap between them to be the longest sequence of one or more consecutive words that occurs in both glosses such that neither the first nor the last word is a function word, that is a pronoun,

[9] `http://www.wiktionary.org/`

[10] We used the *JWKTL* library (Zesch et al., 2008a) to access Wiktionary data of October 2011.

[11] Available in the WordNet::Similarity tool (Pedersen et al., 2004).

> preposition, article or conjunction. ... The sizes of the overlaps thus found are squared and added together to arrive at the score for the given pair of glosses."

The *GlossVectors* (Patwardhan and Pedersen, 2006) measure is calculated as a cosine between vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ representing concepts c_i and c_j. A vector $\mathbf{v}_i$ is a sum of context vectors (derived as in the *BDA* measures) calculated on a corpus of all WordNet glosses. A vector $\mathbf{v}_i$ is a sum of context vectors representing all words from the definition of c_i and the definitions of terms related to c_i:

$$s_{ij} = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{||\mathbf{v}_i||||\mathbf{v}_j||} \text{ where } \mathbf{v}_i = \sum_{\forall j: c_j \in G_i} \mathbf{f}_j. \tag{3.18}$$

Here $\mathbf{f}_j$ is a context vector, derived from the corpus of all glosses and G_i is concatenation of glosses of the concept c_i and all concepts which are directly related to it.

The complexity of the definition-based measures is mainly bounded by the time required to preprocess definitions and calculate pairwise similarities between them. In that respect, measures based on Wiktionary and WordNet are similar since they use the bag-of-word model to represent terms.

The definition-based measures are limited by the number of available definitions. As of October 2011, WordNet contained 117,659 definitions (glosses), Wiktionary contained 536,594 definitions in English and 4,272,902 definitions in all languages and Wikipedia had 3,866,773 English articles and 20.8 million of articles for all languages.

3.5 Classification of the Measures

Figure 3.3 contains a classification of the measures considered in this section. It might help to understand the results of the comparison. For instance, one may expect that the measures from the same group will provide similar results.

In the literature on semantic similarity, some authors emphasize a difference between measures of *semantic similarity* and measures of *semantic relatedness* (Budanitsky and Hirst, 2006). For instance, a network-based measure of semantic similarity should use only hierarchical and equivalence relations of a semantic network, while a measure of semantic relatedness also uses relations of other types (meronyms, associations, etc.). Semantic relatedness is a more general notion than semantic similarity. According to this criterion, *Resnik, Jiang-Conrath, Lin, InvEdgeCount, LeacockChodorow* and *WuPalmer* are measures of semantic similarity, while *ExtendedLesk* and *GlossVectors* are measures of semantic relatedness.

Different measures use different sources of information to compute similarity scores. *InvEdge-*

Count, LeacockChodorow and *WuPalmer* are "pure" network-based measures. On the other hand, *Resnik, JiangConrath,* and *Lin* combine information from a semantic network and a corpus. *ExtendedLesk* and *GlossVectors* rely at the same time on the structure of a semantic network and definitions of terms. *DefVectors-Wkt* and *DefVectors-WktWiki* measures use only definitions, while *BDA, SDA* and *LSA* rely only on a text corpus. Web-based measures *NGD* and *PMI* build only on the top on a Web as a corpus. Distributional measures, web-based measures and *LSA* are calculated differently, but they all rely on co-occurrence of terms in the documents. Contrastingly, *PatternSim* relies on the co-occurrence of terms within specific extraction concordances. Finally, one common feature of *DefVectors, ExtendedLesk, GlossVectors, BDA, SDA* and *LSA* is that all of them rely on the Vector Space Model.

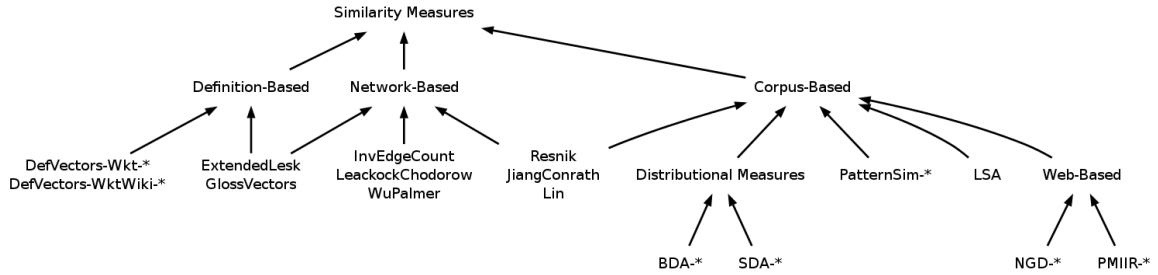


Figure 3.3: Classification of the semantic similarity measures used compared in this section.

3.6 Results

Our comparison of similarity measures is based on human judgments about semantic similarity and on the semantic relation ranking task (see Section 1.2.3).

3.6.1 Correlation with Human Judgments

In this section, we compare semantic similarity measures on the task of correlation with human judgments (see Section 1.2.3). Table 3.1 presents performance of the 35 network-, corpus-, and definition-based similarity measures on this benchmark. We ranked the measures according to their Spearman's correlation. The best measures in each group (network-, corpus-based, etc.) are highlighted in bold.

The correlations of most web-based measures with human judgments are low and not significant in most of the cases. *PMIIR-GoogleWiki* and *NGD-GoogleWiki* which are based on a small Wikipedia corpus are two exceptions. They provided the best results among the web measures. However, all measures perform generally far better than those relying on the Web as a corpus. This has three causes. First, web-based measures rely on the co-occurrence

of terms within a very large context of the whole document. Second, Web pages contain boilerplates, which introduce noise in the results. This is especially true for common words on the Web, such as "flash". Last, but not the least, the number of hits returned by Google and the others is a raw approximation of the real count.

Particularly high correlations with human judgments were observed for the following measures: *Resnik, LeacockChodorow, SDA-*, BDA-3-Cos* and *DefVectors-WktWiki-**. Table 3.2 illustrates rankings obtained with some of these best measures on the MC pairs. As we can see, the pairs "automobile-car", "boy-lad", "gem-jewel" and "journey-voyage" are in the top lists provided by both humans and computer programs. Similarly, the pairs "rooster-voyage", "noon-string" and "cord-smile" are on the bottom of all the lists.

While the MC and RG score high only synonymy-, co-hyponymy-, and hypernymy-like pairs (see Table 3.2), the WordSim scores high associations as well. This is a likely reason why the network-based measures, such as *Resnik*, perform well on the MC and RG and worse on the WordSim. On the other hand, the corpus based measures, such as *LSA-Tasa*, are able to retrieve associations and thus perform better on the WordSim benchmark.

3.6.2 Semantic Relation Ranking

In this section, we compare semantic similarity measures on the task of semantic relation ranking (see Section 1.2.3). Table 3.3 provides an example of such relation ranking performed by the distributional measure measure *BDA-sent-Cos*. We compared performances of 37 network-, corpus-, and definition-based similarity measures on this benchmark. Quality of each similarity measures was quantified with the four following statistics: Precision(10), Precision(20), Precision(50) and Fmeasure(50).

Table 3.4 presents results of this comparison. We ranked the measures according to Precision(20) and Fmeasure(50) statistics. Network-, corpus-, and definition-based measures are grouped and the best measures in each group are in bold. The following single measures provided the best scores in this experiment: *Resnik, WuPalmer, JiangConrath, BDA-3-Cos, SDA-21-Cos, PatternSim-EfreqCfreqRnumPnum* and *DefVectors-WktWiki-**. Figure 3.4 (a) presents precision-recall graph of *Resnik, SDA-21-Cos, BDA-3-Cos* and *DefVectors-WktWiki-1000*. As one may see, definitions yield less precise models than a corpus. *SDA-21-Cos* seems to be the most precise measure, but its recall is lower than that of *BDA* due to the sparsity of the syntactic contexts. On the other hand, recall of *DefVectors* and *Resnik* is even worse as these measures rely on a set of definitions.

Figure 3.4 (b) depicts Precision-Recall graph of four variations of the definition-based measures. Our experiments showed that the measures which use both Wiktionary and Wikipedia

	Sim.Measure	MC Dataset		RG Dataset		WordSim Dataset	
		Pearson	*Spearman*	*Pearson*	*Spearman*	*Pearson*	*Spearman*
	Random	0.172 ***	0.056 ***	-0.060 ***	-0.047 ***	-0.158 ***	-0.122 ***
Network-Based	Resnik	0.823	**0.784**	**0.823**	0.757	0.350	**0.330**
	InvEdgeCount	0.755	0.724	0.782	**0.789**	**0.366**	0.295
	LeackockChodorow	**0.779**	0.724	0.841	**0.789**	0.313	0.295
	WuPalmer	0.768	0.742	0.800	0.775	0.270	0.330
	Lin	0.769	0.754	0.737	0.619	0.287	0.203
	JiangConrath	0.473 *	0.719	0.575	0.587	0.227	0.175
Corpus-Based	BDA-sent-Cos	0.642	0.638	0.694	0.703	0.383	0.362
	BDA-1-Cos	0.658	0.676	0.704	0.758	0.448	0.438
	BDA-2-Cos	0.667	0.638	0.698	0.734	0.441	0.439
	BDA-3-Cos	0.722	0.692	0.752	0.782	0.467	0.465
	BDA-5-Cos	0.710	0.683	**0.755**	0.787	0.467	0.455
	BDA-8-Cos	0.707	0.697	0.746	0.764	0.455	0.440
	BDA-10-Cos	0.710	0.718	0.746	0.764	0.443	0.425
	SDA-6-Cos	0.759	**0.790**	0.741	**0.792**	0.380	0.496
	SDA-9-Cos	0.756	**0.790**	0.732	0.787	0.384	0.491
	SDA-21-Cos	**0.756**	**0.790**	0.731	0.785	0.384	0.490
	NGD-Bing	0.035 ***	0.063 ***	0.174 ***	0.181 ***	0.042 ***	0.058 ***
	NGD-Yahoo	0.387 **	0.330 ***	0.448	0.445	0.290	0.254
	NGD-Google	0.085 ***	0.019 ***	-0.013 ***	-0.012 ***	0.120 **	0.150 *
	NGD-GoogleWiki	0.306 ***	0.334 ***	0.452	0.501	0.205	0.250
	PMIIR-Bing	0.079 ***	0.120 ***	0.116 ***	0.149 ***	0.000 ***	0.003 ***
	PMIIR-Google	0.046 ***	-0.107 ***	-0.061 ***	-0.039 ***	0.097 ***	0.113 **
	PMIIR-GoogleWiki	0.508 *	0.498 *	0.401	0.411	0.254	0.279
	PMIIR-Factiva	0.312 ***	0.442 **	0.436	0.517	0.314	0.559
	NGD-Factiva	0.602	0.602	0.618	0.599	**0.565**	0.599
	PatternSim-Efreq	0.178 ***	0.486	0.150 ***	0.632	0.165 ***	0.430
	PatternSim-EfreqCfreq	0.331 ***	0.600	0.376	0.709	0.191	0.493
	PatternSim-EfreqCfreqRnumPnum	0.271 ***	0.645	0.334 ***	0.733	0.145 ***	0.520
	LSA-Tasa	0.737	0.694	0.645	0.604	0.527	**0.565**
Definition-Based	GlossVectors	0.566	0.653	0.647	0.738	0.383	0.322
	ExtendedLesk	0.355 ***	0.792	0.340 *	0.717	0.209	0.409
	DefVectors-Wkt-1000	0.625	0.687	0.655	0.760	**0.416**	0.492
	DefVectors-Wkt-2500	0.625	0.687	0.655	0.760	0.382	0.527
	DefVectors-WktWiki-1000	**0.704**	**0.759**	**0.701**	**0.754**	0.453	**0.545**
	DefVectors-WktWiki-2500	**0.704**	0.759	**0.701**	0.754	**0.416**	0.520

Table 3.1: Evaluation on the human judgment datasets (MC, RG and WordSim). Here (*) means $p \leq 0.01$, (**) means $p \leq 0.05$, (***) means $p > 0.05$, otherwise $p \leq 0.001$. The best results for each group of measures are in bold. The best result in a column is in grey.

(denoted as *DefVectors-WktWiki*) are better on most of the datasets than the measures relying only on Wiktionary (denoted as *DefVectors-Wkt*). In particular, *DefVectors-WktWiki-1000* outperformed all definition-based measures, including those based on WordNet glosses. On the BLESS dataset, the syntactic distributional measure *SDA-21-Cos* achieved the best precision among the single measures, while bag-of-words distributional analysis *BDA-3-Cos* achieved the highest F-measure. On the SN dataset, the WordNet-based measure *WuPalmer* performed best, achieving a highest precision and a F-measure.

3.6.3 Comparison of Semantic Relation Distributions

In this section, we use the BLESS dataset to perform a study "on the specific aspects of lexical knowledge captured by the models" as suggested by Baroni and Lenci (2011). We try to figure out what types of semantic relations each measure extract. We compare the

Rank	**MC**	**Resnik**	**BDA-3-Cos**	**SDA-21-Cos**	**DefVectors-WktWiki-1000**
1	automobile-car	automobile-car	midday-noon	asylum-madhouse	asylum-madhouse
2	journey-voyage	gem-jewel	automobile-car	automobile-car	automobile-car
3	gem-jewel	journey-voyage	gem-jewel	bird-crane	bird-crane
4	boy-lad	boy-lad	magician-wizard	bird-cock	bird-cock
5	coast-shore	coast-shore	implement-tool	boy-lad	boy-lad
6	asylum-madhouse	asylum-madhouse	journey-voyage	brother-monk	brother-monk
7	magician-wizard	magician-wizard	coast-shore	brother-lad	brother-lad
8	midday-noon	midday-noon	furnace-stove	car-journey	car-journey
9	furnace-stove	furnace-stove	boy-lad	cemetery-woodland	cemetery-woodland
10	food-fruit	food-fruit	food-fruit	coast-shore	coast-shore
11	bird-cock	bird-cock	coast-hill	coast-forest	coast-hill
12	bird-crane	bird-crane	car-journey	coast-hill	coast-forest
13	implement-tool	implement-tool	brother-lad	cord-smile	cord-smile
14	brother-monk	brother-monk	coast-forest	crane-implement	crane-implement
15	crane-implement	brother-lad	brother-monk	food-fruit	food-fruit
16	brother-lad	crane-implement	bird-crane	food-rooster	food-rooster
17	car-journey	car-journey	monk-slave	forest-graveyard	forest-graveyard
18	monk-oracle	monk-oracle	lad-wizard	furnace-stove	furnace-stove
19	cemetery-woodland	cemetery-woodland	forest-graveyard	gem-jewel	gem-jewel
20	food-rooster	food-rooster	shore-woodland	glass-magician	glass-magician
21	coast-hill	coast-hill	cemetery-woodland	implement-tool	implement-tool
22	forest-graveyard	forest-graveyard	food-rooster	journey-voyage	journey-voyage
23	shore-woodland	shore-woodland	glass-magician	lad-wizard	lad-wizard
24	monk-slave	monk-slave	bird-cock	magician-wizard	magician-wizard
25	coast-forest	coast-forest	crane-implement	midday-noon	midday-noon
26	lad-wizard	lad-wizard	monk-oracle	monk-slave	monk-slave
27	cord-smile	cord-smile	asylum-madhouse	monk-oracle	monk-oracle
28	glass-magician	glass-magician	cord-smile	noon-string	noon-string
29	noon-string	rooster-voyage	noon-string	rooster-voyage	rooster-voyage
30	rooster-voyage	noon-string	rooster-voyage	shore-woodland	shore-woodland

Table 3.2: Ranking of the word pairs from the MC dataset by the four best similarity measures.

Rank	**ant**	**banana**	**fork**	**missile**	**salmon**
1	cockroach (cohypo)	mango (cohypo)	prong (mero)	warhead (mero)	trout (cohypo)
2	grasshopper (cohypo)	pineapple (cohypo)	spoon (cohypo)	weapon (hyper)	mackerel (cohypo)
3	silverfish (cohypo)	papaya (cohypo)	knife (cohypo)	deploy (event)	herring (cohypo)
4	wasp (cohypo)	pear (cohypo)	lift (event)	nuclear (attri)	fish (event)
5	insect (hyper)	ripe (attri)	**fender (random)**	bomb (cohypo)	tuna (cohypo)
6	arthropod (hyper)	peach (cohypo)	plate (cohypo)	destroy (event)	oily (attri)
7	industrious (attri)	coconut (cohypo)	rake (cohypo)	rocket (cohypo)	poach (event)
8	ladybug (cohypo)	fruit (hyper)	shovel (cohypo)	arm (hyper)	catfish (cohypo)
9	bee (cohypo)	apple (cohypo)	handle (mero)	propellant (mero)	catch (event)
10	beetle (cohypo)	apricot (cohypo)	sharp (attri)	**bolster (random)**	fresh (attri)
11	locust (cohypo)	strawberry (cohypo)	spade (cohypo)	launch (event)	cook (event)
12	dragonfly (cohypo)	ripen (event)	napkin (cohypo)	deadly (attri)	cod (cohypo)
13	hornet (cohypo)	plum (cohypo)	cutlery (hyper)	**country (random)**	smoke (event)
14	creature (hyper)	grapefruit (cohypo)	head (mero)	strike (event)	seafood (hyper)
15	crawl (event)	cherry (cohypo)	scissors (cohypo)	defuse (event)	eat (event)

Table 3.3: Examples of the semantic relation ranking with the measure BDA-sent-Cos. The word in braces denote a relation type from the BLESS dataset.

distributions of 21 semantic measures with help of the BLESS dataset. If two measures have comparable general performances, one may want to choose a measure which provides more relations of a certain type, depending on the application. This information may be also valuable to decide which measures to combine in a meta-measure.

	Sim.Measure	BLESS Dataset				SN Dataset			
		P(10)	*P (20)*	*P(50)*	*F(50)*	*P(10)*	*P(20)*	*P(50)*	*F(50)*
	Random	0.546	0.541	0.543	0.522	0.504	0.501	0.498	0.498
Network-Based	Resnik	0.977	**0.958**	0.718	0.690	0.948	0.908	0.725	0.725
	InvEdgeCount	0.967	0.925	0.722	0.693	0.981	0.947	0.752	0.752
	LeackockChodorow	0.967	0.925	0.722	0.693	**0.982**	0.951	0.756	0.756
	WuPalmer	0.978	0.938	0.706	0.678	0.979	**0.959**	**0.764**	**0.764**
	Lin	0.975	0.919	0.776	0.745	0.924	0.853	0.637	0.637
	JiangConrath	**0.981**	0.909	**0.732**	**0.703**	0.916	0.835	0.615	0.615
Corpus-Based	BDA-sent-Cos	0.962	0.920	0.799	0.767	0.941	0.898	0.724	0.724
	BDA-1-Cos	0.971	0.940	0.826	0.793	0.969	0.926	0.737	0.737
	BDA-2-Cos	0.966	0.939	0.829	0.796	0.970	0.929	0.738	0.738
	BDA-3-Cos	0.970	0.947	0.835	**0.802**	0.974	0.932	0.743	0.743
	BDA-5-Cos	0.975	0.946	0.833	0.800	0.971	0.929	0.744	0.744
	BDA-8-Cos	0.974	0.943	0.827	0.794	0.968	0.924	0.741	0.741
	BDA-10-Cos	0.972	0.941	0.821	0.789	0.962	0.922	0.737	0.737
	SDA-6-Cos	0.984	0.948	0.810	0.778	0.978	0.945	0.749	0.749
	SDA-9-Cos	0.984	0.951	0.809	0.777	**0.977**	0.945	0.753	**0.753**
	SDA-21-Cos	0.985	0.953	0.810	0.778	0.978	**0.946**	0.753	**0.753**
	NGD-Bing	0.725	0.692	0.695	0.670	0.676	0.682	0.639	0.639
	NGD-Yahoo	0.940	0.907	0.782	0.751	—	—	—	—
	NGD-YahooBoss	0.847	0.843	0.747	0.718	—	—	—	—
	NGD-Google	0.991	0.934	0.651	0.625	—	—	—	—
	NGD-GoogleWiki	0.874	0.836	0.702	0.674	—	—	—	—
	PMIIR-Bing	0.675	0.650	0.692	0.667	0.610	0.608	0.647	0.647
	PMIIR-YahooBoss	0.823	0.822	0.724	0.696	—	—	—	—
	PMIIR-Google	0.822	0.749	0.660	0.634	—	—	—	—
	PMIIR-GoogleWiki	0.791	0.761	0.676	0.649	—	—	—	—
	NGD-Factiva	0.959	0.916	0.800	0.768	0.900	0.832	0.651	0.651
	PMIIR-Factiva	0.903	0.860	0.816	0.784	0.826	0.768	0.606	0.606
	PatternSim-Efreq	0.980	0.945	**0.909**	0.544	0.938	0.915	0.866	0.547
	PatternSim-EfreqCfreq	**0.989**	0.956	**0.909**	0.544	0.949	0.920	**0.867**	0.547
	PatternSim-EfreqCfreqRnumPnum	**0.989**	**0.957**	**0.909**	0.544	0.952	0.924	**0.867**	0.547
	LSA-Tasa	0.967	0.936	0.801	0.769	0.901	0.839	0.637	0.637
Definition-Based	GlossVectors	0.894	0.860	0.742	0.712	0.930	0.872	0.719	0.719
	ExtendedLesk	0.940	0.870	0.716	0.687	**0.950**	0.895	0.653	0.653
	DefVectors-Wkt-1000	0.926	0.885	0.783	0.752	0.907	0.868	0.678	0.678
	DefVectors-Wkt-2500	0.915	0.882	0.754	**0.754**	0.928	0.898	0.704	0.704
	DefVectors-WktWiki-1000	**0.942**	**0.905**	**0.785**	0.725	0.917	0.878	0.696	0.696
	DefVectors-WktWiki-2500	0.931	0.891	0.765	0.734	0.937	**0.912**	**0.726**	**0.726**

Table 3.4: Evaluation of the measures on the semantic relation datasets (BLESS and SN). Here P(x) and F(x) are Precision and F-measure as specified in Section 1.2.3. The best results for each column are highlighted with gray color.

Distribution of Relation Types

In this section, we estimate empirical relation distribution of the measures over five relation types: hypernymy, co-hyponymy, meronymy, attribute and event. To do so, we calculate percentage of correctly extracted relations of type t for each measure:

$$Percent = \frac{\hat{R}_t}{|R \cap \hat{R}|} \text{ where } \bigcup_{t \in T} \hat{R}_t = |R \cap \hat{R}|. \tag{3.19}$$

Here $|R \cap \hat{R}|$ is a set of all correctly extracted relations and $\hat{R}_t$ is the set of extracted relations of type t. Figure 3.5 demonstrates that the percent of extracted relations of a certain type depends on the value of k (c. f. Section 1.2.3). For instance, if k equals 10%, then 77% of extracted relations by Resnik are co-hyponyms, but if k equals 40%, then the same

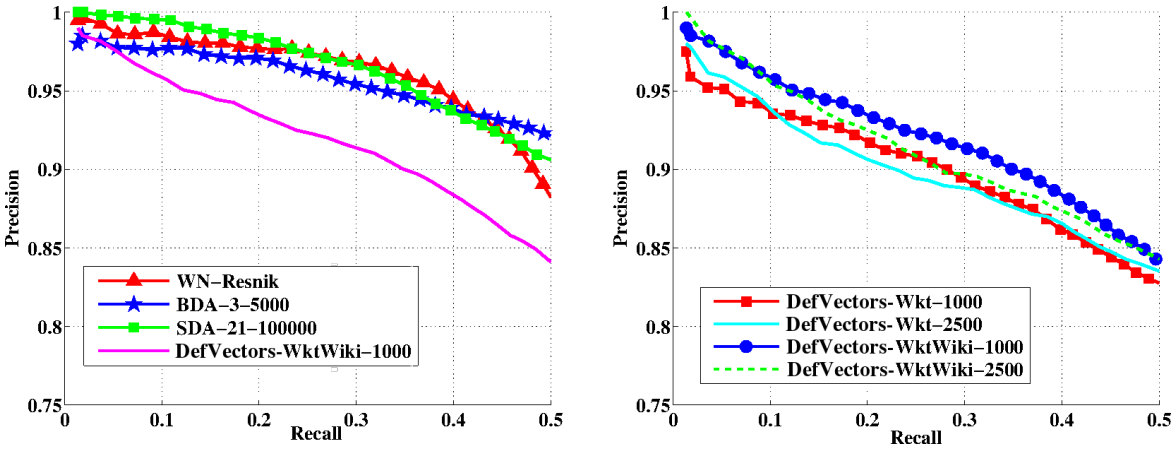


Figure 3.4: Precision-Recall graphs of (a) four best similarity measures; (b) similarity measures based on definitions of Wiktionary and Wikipedia.

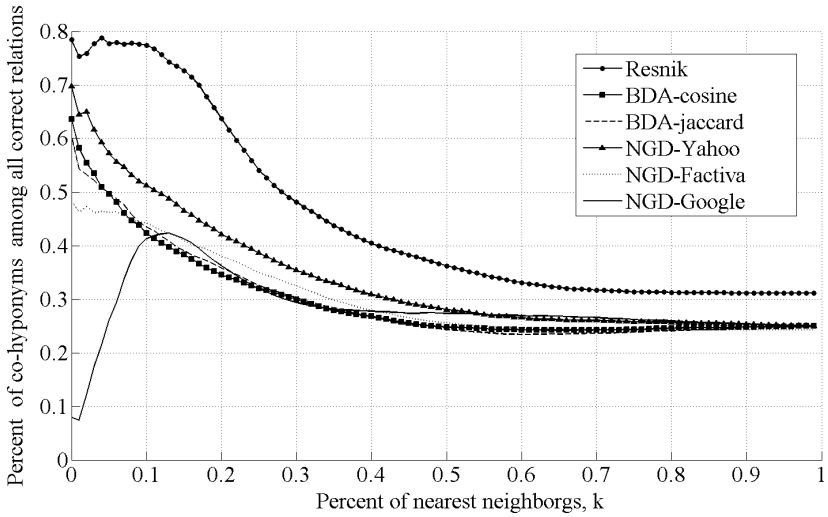


Figure 3.5: Percentage of co-hyponyms among correctly extracted relations for the six best measures.

measure outputs 40% of co-hyponyms. We report the relation distributions at two levels of the threshold k – 10% and 40%. The empirical distributions are reported in Table 3.5. Each column corresponds to one semantic relation type t and contains two numbers:

- $Percent(10)$ – percent of relations of type t for $k = 10\%$,
- $Percent(40)$ – percent of relations of type t for $k = 40\%$.

We represent those two values in the following format: $Percent(10)|Percent(40)$. For instance, $77|40$ means that when $k = 10\%$, it extracts 77% of co-hyponyms and when $k = 40\%$, it extracts 40% of co-hyponyms.

If the threshold k is 10%, then the biggest fraction of extracted relations are co-hyponyms – from 35% for *BDA-sent-Manhattan* to 77% for *Resnik* measure. At this threshold level, the network-based measures mostly return co-hyponyms (60% in average) and hypernyms (23% in average). The corpus-based measures mostly return co-hyponyms (38% in average) and event relations (26% in average). The web-based measures return many (48% in average) co-hyponymy relations.

If the threshold k is 40%, then relation distribution for each measure significantly changes. Most of the relations returned by the network-based measures are co-hyponyms (36%) and meronyms (24%). The majority of relations discovered by the corpus-based measures are co-hyponyms (33%), event relations (26%) and meronyms (20.33%). The web-based measures at this threshold value return many event relations (32%).

Interestingly, for most of the measures, the percent of extracted hypernyms and co-hyponyms decreases as the value of k increases, while the percent of other relations increases. To make it clear, we grayed cells of the Table 3.5 when $Percent(10) \geq Percent(40)$.

Similarity to the BLESS Distribution

In this section, we would like to make sure that the relation distributions (see Table 3.5) are not biased by the distribution of the evaluation dataset. We compare relation distributions of the measures with the distribution in the BLESS on the basis of the χ^2 goodness of fit test with $df = 4$ (Agresti, 2002) [12]. A random similarity measure appeared to be biased by the distribution in the evaluation dataset: $\chi^2 = 5.36$, $p = 0.252$ for $k = 10\%$ and $\chi^2 = 3.17$, $p = 0.53$ for $k = 40\%$. On the other hand, distributions of all the 21 measures are significantly different from the distribution in the BLESS ($p < 0.001$). The value of the chi-square statistic varies from $\chi^2 = 89.94$ (*NGD-Factiva*, $k = 10\%$) to $\chi^2 = 4000$ (*Resnik*, $k = 10\%$). The distributions are more similar to the BLESS at the threshold level $k = 40\%$ ($\chi^2 = 868$ in average for all measures) with respect to the threshold value $k = 10\%$ ($\chi^2 = 1467$ in average).

Independence of Relation Distributions

In this section, we check whether relation distributions of the various measures are significantly different. To do so, we performed the chi-square independence test on the Table 3.5. Our experiments have shown that there is a significant interaction between the type of the measure and the relations distribution: $\chi^2(80) = 10487, p < 0.001$ for all the measures; $\chi^2(28) = 2529, p < 0.001$ for the network-based measures; $\chi^2(12) = 245, p < 0.001$ for the corpus-based measures; $\chi^2(32) = 3158, p < 0.001$ for the web-based measures. Thus, there is a clear dependence between the type of the measure and the type of the relation it extracts. We found that relation distributions in the four groups listed above are more similar (values of the chi-square statistic are lower) for the threshold $k = 40\%$ with respect to $k = 10\%$.

[12] Here and below, we calculate the χ^2 statistic from the Table 3.5 (columns 5-9), where percents are replaced with absolute frequencies. We compare rows of the table ($df = 4$).

	Measure	**hyper**, %	**cohypo**, %	**attri**, %	**mero**, %	**event**, %
Network-Based	Resnik	9 | 14	77 | 40	4 | 8	6 | 22	4 | 15
	InvEdgeCounts	22 | 15	61 | 40	4 | 8	7 | 22	6 | 15
	LeacockChodorow	22 | 15	61 | 40	4 | 8	7 | 22	6 | 15
	WuPalmer	20 | 15	64 | 42	3 | 8	7 | 22	5 | 13
	Lin	30 | 16	52 | 31	4 | 7	8 | 29	5 | 16
	GlossOverlap	5 | 6	52 | 34	7 | 12	18 | 21	18 | 27
	JiangConrath	38 | 16	45 | 30	4 | 6	8 | 29	5 | 18
	Extended Lesk	21 | 14	39 | 30	1 | 9	29 | 28	9 | 19
Corpus-Based	BDA-sent-Cos	9 | 7	42 | 27	11 | 20	15 | 17	23 | 30
	BDA-sent-Jaccard	10 | 7	45 | 27	8 | 16	16 | 20	20 | 27
	BDA-sent-Manhattan	7 | 6	35 | 24	17 | 22	10 | 15	31 | 34
	BDA-sent-Euclidean	7 | 7	31 | 18	20 | 26	12 | 13	30 | 37
	NGD-Yahoo	7 | 6	51 | 30	9 | 18	17 | 20	15 | 25
	NGD-Factiva	10 | 8	44 | 28	8 | 19	23 | 22	16 | 25
	NGD-YahooBoss	13 | 10	54 | 36	4 | 10	14 | 20	15 | 22
	NGD-Google	1 | 7	41 | 28	45 | 19	2 | 19	11 | 28
	NGD-GoogleWiki	8 | 9	45 | 31	8 | 14	20 | 21	19 | 25
	PMIIR-YahooBoss	15 | 12	53 | 38	3 | 9	15 | 20	13 | 20
	PMIIR-Factiva	8 | 8	42 | 30	10 | 17	21 | 20	18 | 24
	PMIIR-Google	8 | 8	55 | 35	7 | 15	17 | 21	12 | 22
	PMIIR-GoogleWiki	12 | 11	47 | 38	7 | 11	20 | 20	13 | 19
	Random	8 | 9	24 | 25	20 | 19	22 | 20	26 | 27
	BLESS dataset	9	25	20	19	27

Table 3.5: Percent of relations of a certain type with respect to all correct relations at the level of k-NN threshold k of 10% or 40%. Notation: $Percent(10)|Percent(40)$.

Most Similar Measures			**Most Dissimilar Measures**		
sim_i	sim_j	x_{ij}	sim_i	sim_j	x_{ij}
LeacockChodorow	InvEdgeCounts	0	NGD-Google	ExtendedLesk	39935
BDA-sent-Jaccard	BDA-sent-Cos	7	JiangConrath	NGD-Google	27479
NGD-YahooBoss	PMIIR-YahooBoss	20	Lin	NGD-Google	17527
WuPalmer	InvEdgeCounts	24	NGD-Google	WuPalmer	17417
WuPalmer	LeacockChodorow	24	NGD-Google	PMIIR-YahooBoss	13391
BDA-sent-Manhattan	BDA-sent-Euclidean	25	InvEdgeCounts	NGD-Google	12013
PMIIR-GoogleWiki	NGD-Factiva	28	LeacockChodorow	NGD-Google	12013
PMIIR-Google	NGD-Yahoo	33	NGD-Google	Resnik	11750
NGD-GoogleWiki	NGD-Factiva	40	NGD-Google	NGD-YahooBoss	11557
NGD-GoogleWiki	PMIIR-Factiva	42	BDA-sent-Euclidean	ExtendedLesk	8412
GlossOverlap	NGD-Yahoo	54	NGD-Factiva	NGD-Google	8067
NGD-Factiva	PMIIR-Factiva	58	BDA-sent-Euclidean	Resnik	6830
Lin	JiangConrath	59	PMIIR-GoogleWiki	NGD-Google	6575
GlossOverlap	NGD-GoogleWiki	62	BDA-sent-Manhattan	ExtendedLesk	6428

Table 3.6: List of the most and most dissimilar measures ($k = 10\%$).

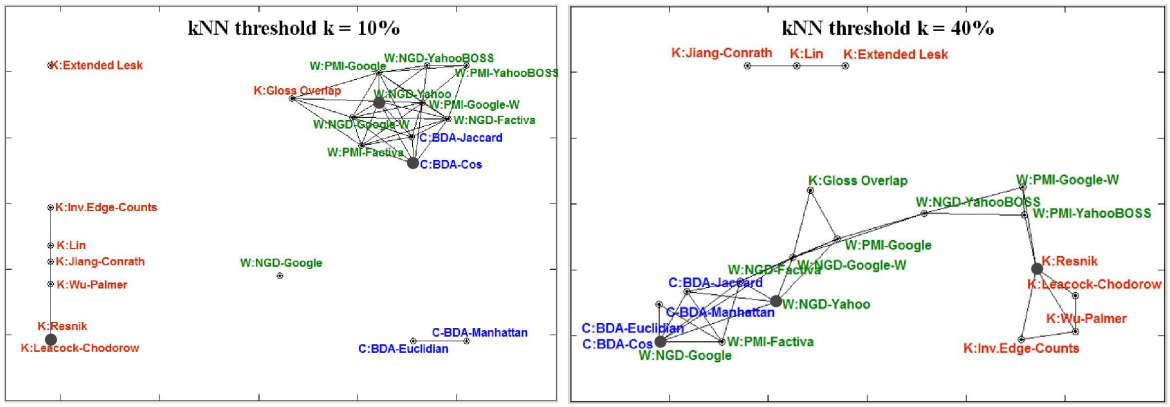


Figure 3.6: 21 semantic similarity measures grouped according to similarity of their relation distributions with the formula (3.20). An edge links measures sim_i and sim_j if $x_{ij} < 220$. The network-, corpus-, and web-based measures are marked in red, blue and green correspondingly and with the prefixes 'K','C' and 'W'. The best measures are marked with a big circle.

Most Similar and Dissimilar Measures

In this section, we would like to find the most similar and dissimilar measures. This information is particularly useful for the combination of the measures. To find redundant measures, we calculate distance x_{ij} between measures sim_i and sim_j with the χ^2-statistic:

$$x_{ij} = x_{ji} = \sum_{t \in T} \frac{(|\hat{R}_t^i| - |\hat{R}_t^j|)^2}{|\hat{R}_t|}, \tag{3.20}$$

where $\hat{R}_t^i$ is a set of correctly extracted relations of the type t with the measure sim_i and $\hat{R}_i$ is a set of extracted relations of type t. We calculate these distances for all pairs of measures and then rank the pairs according to the value of x_{ij}. Table 3.6 presents a list of the most similar and dissimilar measures obtained this way. Figure 3.6 reports in a compact way all the pairwise similarities $(x_{ij})_{21\times21}$ between the 21 measures. In this graph, an edge links two measures, which have the distance value $x_{ij} < 220$. The graph was drawn with the Fruchterman and Reingold (1991) force-directed layout algorithm. One can see that relation distributions of the web- and corpus-based measures are quite similar. The network-based measures are more different from them, but similar among themselves. Figure 3.6 suggests once again that relation distributions are more similar at the threshold level $k = 40\%$ with respect to the threshold level $k = 10\%$.

While Figure 3.3 is a theoretical classification of the measures, Figure 3.6 can be considered as an empirical classification. Interestingly, these two figures group measures in a similar way. For instance, most of network-based measures belong to the same cluster and most corpus-based measures belong to the same cluster as well.

Distribution of Similarity Scores

In this section, we compare distributions of similarity scores across relation types to identify the dominant relation types. We rely on the following procedure proposed by Baroni and Lenci (2011):

1. Pick a closest relatum concept c_j per relation type t for each target concept c_i.
2. Convert similarity scores associated to each target concept to z-scores.
3. Summarize the distribution of similarities across relations by plotting the z-scores grouped by relations in a box plot.
4. Verify the statistical significance of the differences in similarity scores across relations by performing the Tukey's HSD test.

Figure 3.9 presents the distributions of similarities across various relation types for *Resnik*, *BDA-sent-Cos* and *NGD-Yahoo*. First, meaningful relation types for these three measures are significantly different from random relations ($p < 0.001$). The only exception is the *Resnik* measure. Its similarity scores for the attribute relations are not significantly different from random relations ($p = 0.178$). Thus, the best three measures provide scores which let us separate incorrect relations from the correct ones if an appropriate threshold k is set. Second, the similarity scores have highest values for the co-hyponymy relations. Third, *BDA-sent-Cos*, *BDA-sent-Jaccard*, *NGD-Yahoo*, *NGD-Factiva* and *PMIIR-YahooBoss* provide the best scores. They let us clearly separate meaningful relations from the random ones ($p < 0.001$). On the other hand, the poorest scores were provided by *BDA-sent-Manhattan*, *BDA-sent-Euclidean*, *NGD-YahooBoss* and *NGD-Google*, because these scores let us clearly separate only co-hyponyms from the random relations.

Corpus Size

Figure 3.7 depicts a learning curve of the *BDA-sent-Cos* measure. Dependence of the F-measure from the training corpus size is not linear. F-measure improves up to 44% when we increase corpus size from 1M to 10M tokens. Increasing corpus from 10M to 100M tokens gives the improvement of 16%. Finally, increasing corpus from 100M to 2000M tokens gives the improvement of only 3%. These and the similar experiments such as (Agirre et al., 2009) suggest that the further linear improvements in the precision would require an exponential growth of the corpus. This motivates the need for more sophisticated corpus-based techniques such as those based on the dependency parsing (Section 2.2, 3.3.1) or the lexico-syntactic patterns (Section 2.4).

Table 3.5 presents the relation distribution of the *BDA-sent-Cos* trained on the 2,000M token corpus *ukWaC*. Figure 3.8 illustrates how relation distribution of this measure depends on

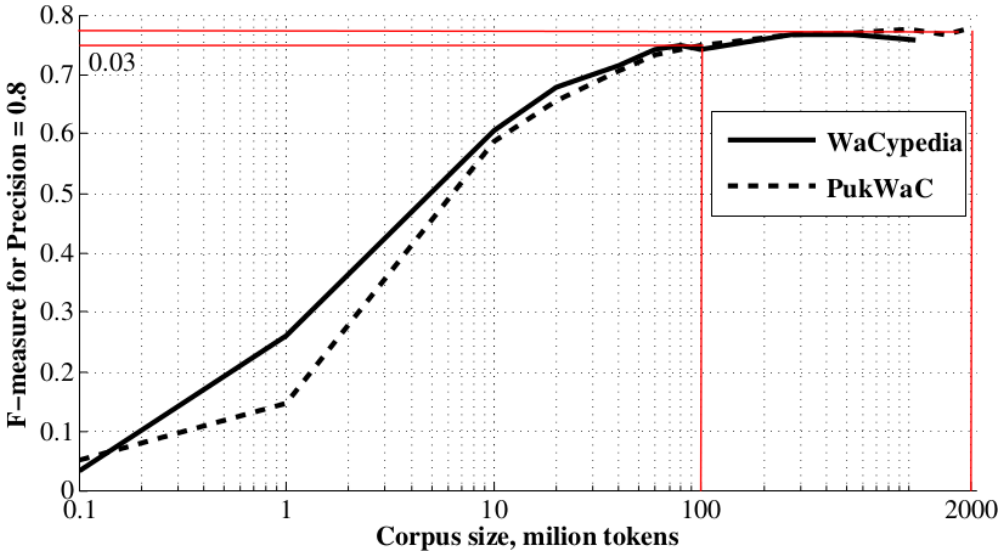


Figure 3.7: Learning curves of the BDA-Cos on the WaCky and PukWaC corpora.

corpus size. First, if corpus size increases, then the percent of attribute relations decreases, while percent of co-hyponyms increases. Second, corpus size does not drastically influence the distribution for big corpora (starting from 100M of tokens). For instance, if we increase corpus size from 100M to 2,000M tokens, then the percent of relations changes on 3% for attributes, 3% for co-hyponyms, 1% for events, 0.7% for hypernyms and on 0.4% for meronyms.

The proportion of the co-hyponymy relations grows on 10% when corpus size grows from 10M to 2000M tokens. Distribution of the other relation types varies less (5% for attribute, 4% for event, 1% for hypernymy and 0.7% for meronymy relations).

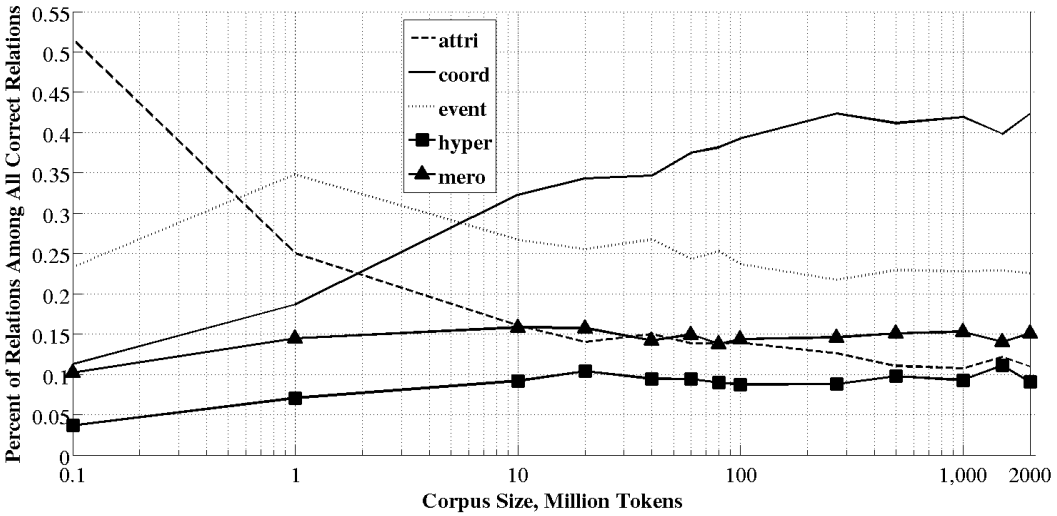


Figure 3.8: Semantic relation distribution function of corpus size (BDA-sent-Cos measure, PukWaC corpus).

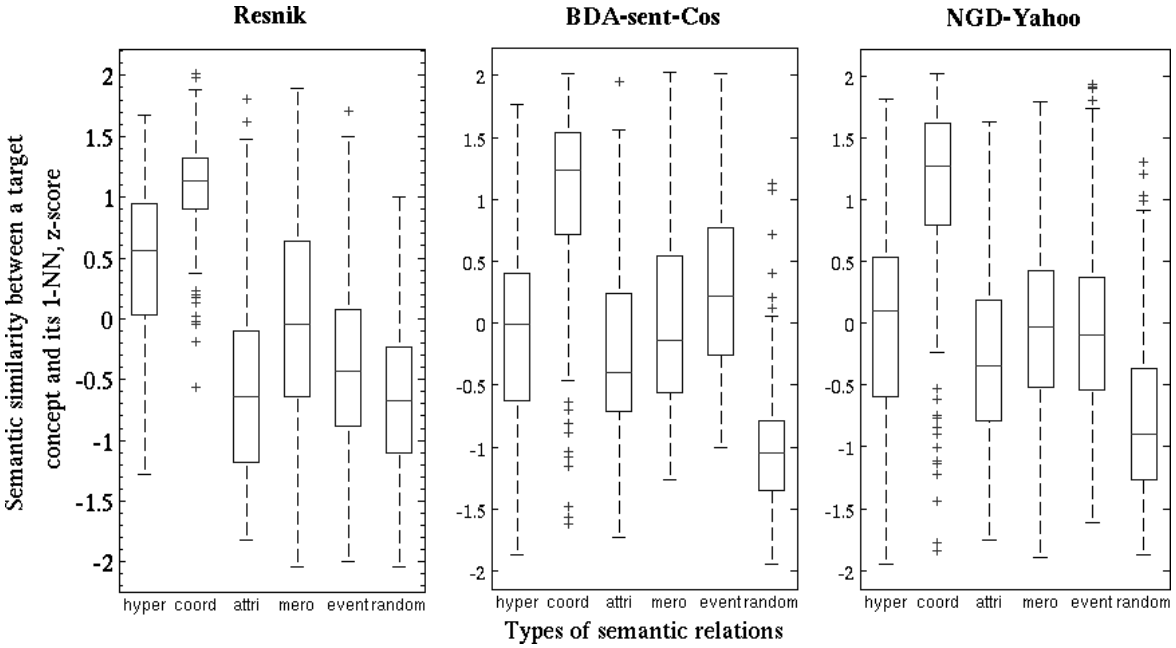


Figure 3.9: Distribution of similarities across relation types for Resnik, BDA-Cos and NGD-Yahoo.

3.7 Discussion

Results obtained on the tasks of correlation with human judgments and semantic relation ranking are overlapping but not identical. We used the following criterion in order to decide which measures are the best: a measure should be the best in its group (e. g., among corpus-based measures) in both types of evaluations. According to this criterion, the best measures are the WordNet measure *Resnik*, the bag-of-words distributional measure *BDA-3-Cos*, the syntactic distributional measure *SDA-21-Cos* and the measure *DefVectors-WktWiki* based on Wiktionary and Wikipedia. Figure 3.10 depicts distributions of similarity scores for these four most successful measures. Our experiments showed that for these measures, there are significant differences in the distributions of scores for meaningful and random relations. This means that an appropriate k-NN threshold level k clearly separates meaningful relations from the random ones.

There is a huge difference in performance between web-based and distributional corpus-based measures. This is likely to be due to the noisy nature of the web documents (*BDA/SDA* use a more precise and linguistically motivated representation of a term) and the fact that the counts of a search engine API are rough approximations of the real counts. Similarly, the higher performance of the network- and definition-based methods is likely due to the more linguistically precise representation of the terms. Some web measures yield significantly worst results than others. Following Veksler et al. (2008), we suggest that this variance in the results is due to differences in the corpora indexed by different search engines. For instance, Web measures over Wikipedia or Factiva corpora provide better results since these

corpora contain less noisy documents than the Web documents indexed by Bing.

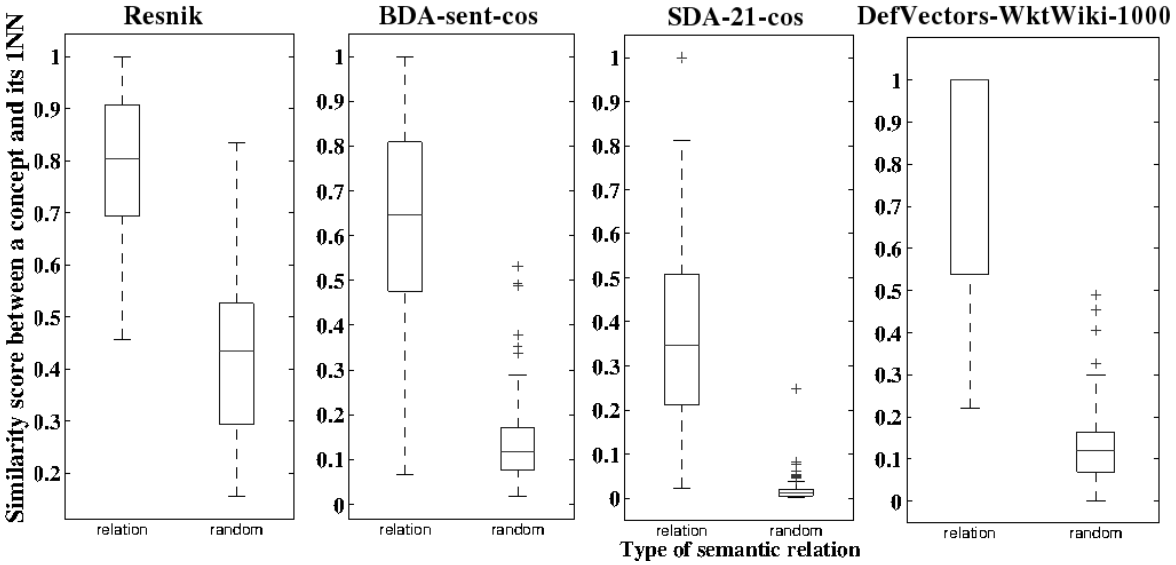


Figure 3.10: Distribution of 1-NN similarity scores of the four best similarity measures on the BLESS dataset. Here "random" and "relation" are distributions of random and meaningful relations.

Prior research provides us some information about general performances of the measures considered in this section. For instance, Mihalcea et al. (2006) compare two corpus-based (*PMI* and *LSA*) and six network-based measures on the task of text similarity computation. The authors report that *PMI* is the best measure; that, similarly to our results, *Resnik* is the best network-based measure; and that simple average over all 8 measures is even better than *PMI*. Budanitsky and Hirst (2006) report that *JiangConrath* is the best network-based measure for the task of spelling correction. Patwardhan and Pedersen (2006) evaluate six network-based measures on the task of word sense disambiguation and report the same result. This contradicts our results, since we found *Resnik* to be the best network-based measure.

Heylen et al. (2008) compared general performances and relation distributions of distributional methods using a lexical database. Sahlgren (2006) evaluated syntagmatic and paradigmatic bag-of-word models. Our findings mostly fits well these and other results on the distributional analysis (e. g. Curran (2003) or Bullinaria and Levy (2007)). Lindsey et al. (2007) compared web-based measures. Authors suggest that a small search domain is better than the whole Web. Our results partially confirm this observation (*NGD-Factiva* outperforms *NGD-Google*) and partially contradicts it (*NGD-Yahoo* outperforms *NGD-Factiva*). Van de Cruys (2010) evaluated BDA and SDA measures and suggests that the syntactic models are the better for the extraction of synonym-like similarities. Wandmacher (2005) reports that *LSA* produces 46.4% of associative relations, 15.2% of synonyms, antonyms, hypernyms, co-hyponyms and meronyms, 5.6% of syntactic relations and 32.8% of erroneous relations.

3.8 Conclusion

In this section, we presented a large-scale comparative study of network-, corpus-, and definition-based semantic similarity measures on several benchmark datasets (MC, RG, WordSim, BLESS and SN). These experiments showed that the *BDA-3-Cos* and the *SDA-21-Cos* measures provided the best performance among corpus-based measures. The measure of *Resnik* performed best among the network-based measures. Finally, the *DefVectors-WktWiki* scored best in the group of definition-based measures. All these measures provide the scores which let us clearly separate correct relations from the random ones.

We also found that semantic relation distributions of different measures are significantly different. However, all measures extract many co-hyponyms. Numerous experiments described in this chapter, suggest that the studied measures are highly heterogeneous in terms of their lexical coverage, performances and semantic relation distributions. There is no single measure which outperforms all others on all benchmarks. We conclude that a multifactor model is required to overcome limitations of the single similarity measures. We address the problem by developing a hybrid semantic similarity measure in the next chapter.

Chapter 4

Hybrid Semantic Similarity Measures[1]

L'union fait la force / Eendracht maakt macht

– the national motto of Belgium

This chapter describes several novel hybrid semantic similarity measures that combine evidence from different sources. We study various unsupervised and supervised combinations of the single similarity measures and evaluate them on the tasks of correlation with human judgments and semantic relation ranking (see Section 1.2.3).

As it was noticed in Sections 2.1 and Chapter 3, the four common approaches to semantic similarity are based respectively on semantic networks, text corpora, Web as a corpus and definitions of dictionaries or encyclopedias. These existing single-resource measures are far from being perfect (see Chapters 2 and 3). To improve the performance, some attempts were made to combine single measures by Curran (2002), Cederberg and Widdows (2003), Mihalcea et al. (2006), Agirre et al. (2009) and Yang and Callan (2009). However, most approaches are still not taking into account the whole range of existing measures, combining mostly sporadically different methods.

The contribution of this chapter are the two-fold:

- First, a systematic analysis of combinations of 16 baseline measures with 9 fusion methods and 3 techniques for the measure selection. We are first to propose hybrid similarity measures based on all main types of resources – corpora, Web corpus, semantic networks, dictionaries and encyclopedias.
- Second, hybrid supervised similarity measures, which combine 15 baseline measures:

[1]The research presented in this chapter has been published as Panchenko and Morozova [6], Panchenko [8] and Panchenko [12].

Logit-E15, *C-SVM-linear-E15*, *C-SVM-radial-E15*, etc. They outperforms all tested single and combined measures by a large margin.

4.1 Features: Single Semantic Similarity Measures

In contrast to the single similarity measures relying on a single source of information, such as those described in Chapters 2 and 3, measures described in this chapter rely on several sources of information at the same time. A *hybrid similarity measure* combines several *single similarity measures* with a *combination method* to achieve better results. In our approach, single similarity measures are features of a hybrid similarity measure (see Figure 4.1). Notice that a hybrid similarity measure has exactly the same inputs and outputs as any single similarity measure. Thus, it is straightforward to apply this kind of measures to the tasks of correlation with human judgments, semantic relation ranking and semantic relation extraction, etc.

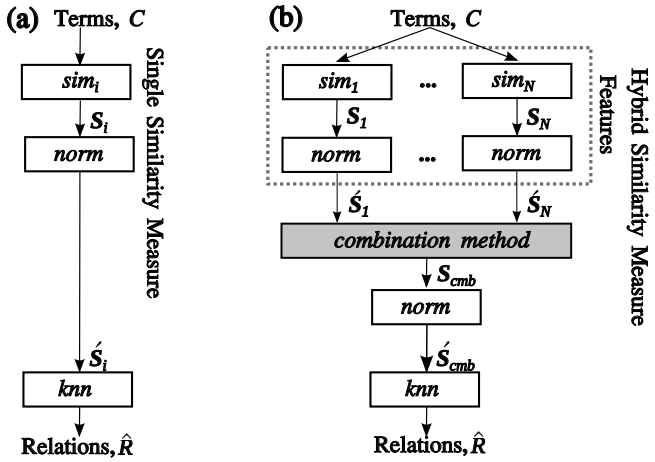


Figure 4.1: (a) Single and (b) hybrid relation extractors based on similarity measures.

In this section, we list 16 baseline measures exploited by the hybrid measures. The single measures were selected as (a) the previous findings suggests that they are able to capture synonyms, hypernyms and co-hyponyms; (b) as they rely on complementary resources to derive semantic similarity – semantic networks, text corpora, Web as a corpus, dictionaries and encyclopedia.

We test five measures relying on *WordNet* semantic network (Miller, 1995b). These measures exploit the lengths of the shortest paths between terms in a network and probability of terms derived from a corpus, as explained in Section 3.2:

1. *WuPalmer* (Wu and Palmer, 1994);
2. *LeackockChodorow* (Leacock and Chodorow, 1998);
3. *Resnik* (Resnik, 1995);

4. *JiangConrath* (Jiang and Conrath, 1997);
5. *Lin* (Lin, 1998a).

We use three measures based on Web as a corpus. These measures use Web search engines for calculation of similarities (see Section 3.3):

6. *NGD-Yahoo* based on the index of *Yahoo!*;
7. *NGD-Bing* based on the index of *Bing*;
8. *NGD-GoogleWiki* based on the index of *Google* over the domain `wikipedia.org`.

We use five measures relying on text corpora to calculate similarity of terms:

9. *BDA-3-Cos*;
10. *SDA-21-Cos*;
11. *PatternSim-Efreq*;
12. *LSA-Tasa*;
13. *NGD-Factiva*.

The 9-th and 10-th measures are based on the distributional analysis of the *WaCky* corpus (see Section 3.3.1). The 11-th measure relies on lexico-syntactic patterns applied to the same corpus (see Section 2.4). The 12-th measure relies on the Latent Semantic Analysis (*LSA*) trained on the TASA corpus, as described in Section 3.3.3. The 13-th measure relies on the *NGD* formula and the Factiva corpus (see Section 3.3.2).

We use three measures that rely on explicit definitions of terms (see Section 3.4 for details):

14. *DefVectors-WktWiki-1000*;
15. *GlossVectors*;
16. *ExtendedLesk*.

The 14-th measure relies on definitions of Wiktionary and Wikipedia and described in detail in Section 3.4. The 15-th and 16-th measure operate rely on WordNet glosses (see Section 3.2).

Different similarity measures listed above are complementary in their coverage. Network-based measures can calculate similarities upon 155,287 English terms from WordNet 3.0. Coverage of web-based measures is huge – it contains all words from all documents indexed by a web search engine. As of July 2012, Google indexed around 50 billion pages, Bing indexed around 17 billion pages, and Yahoo indexed around 3.5 billion pages [2]. Lexical

[2] According to the statistics of `http://www.worldwidewebsize.com/`

coverage of the definition-based measures is limited by the number of available definitions. As of October 2011, WordNet contains 117,659 definitions (glosses); Wiktionary contains 536,594 definitions in English and 4,272,902 definitions in all languages; Wikipedia has 3,866,773 English articles and around 20.8 millions of articles in all languages. Extraction capabilities of these measures are limited by a corpus. For instance, *WaCky* corpus contains 3,368,147 distinct lemmas and *ukWaC* corpus contains 5,469,313 distinct lemmas.

4.2 Combination Methods

A hybrid similarity measure combines several single similarity measures described above with one of the combination methods described below. The goal of a combination method is to produce similarity scores which perform better than the scores of input single measures. A combination method takes as an input a set of similarity matrices $\{\mathbf{S}_1, \ldots, \mathbf{S}_K\}$ produced by K single measures and outputs a combined similarity matrix $\mathbf{S}_{cmb}$. We denote as s_{ij}^k a *pairwise similarity score* of terms c_i and c_j produced by k-th measure. Refer to Kuncheva (2007) for an in-depth overview of the combination methods including the techniques for fusion of the output labels, the techniques for fusion of the continuous-valued outputs and the methods for classifier selection. Some further information about the combination methods can be found in Xu et al. (1992), Kittler (1998), Tax et al. (2000) and (Bishop et al., 2006, p.653).

In this section, we apply several unsupervised and supervised combination methods to our problem. The unsupervised methods all rely on a mean of the scores, e. g. a mean of z-scores, a mean of ranks, etc. The supervised methods rely on a weighted combination of the scores, where weights are learnt automatically.

Mean

This unsupervised combination method simply calculates a mean of K pairwise similarity scores:

$$\mathbf{S}_{cmb} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{S}_k \Leftrightarrow s_{ij}^{cmb} = \frac{1}{K}\sum_{k=1}^{K}s_{ij}^k. \tag{4.1}$$

MeanNnz

This unsupervised combination method calculates a mean of those pairwise similarity scores which have a non-zero value:

$$s_{ij}^{cmb} = \frac{1}{|k : s_{ij}^{k} > 0, k = 1, \ldots, K|} \sum_{k=1}^{K} s_{ij}^{k}. \tag{4.2}$$

MeanZscore

This unsupervised combination method calculates a mean of K similarity scores transformed into Z-scores:

$$s_{ij}^{cmb} = \frac{1}{K} \sum_{k=1}^{K} \frac{s_{ij}^{k} - \mu_k}{\sigma_k}, \tag{4.3}$$

where μ_k is a mean and σ_k is a standard deviation of similarity scores of k-th measure ($\mathbf{S}_k$).

Median

This unsupervised combination method calculates a median of K pairwise similarities:

$$s_{ij}^{cmb} = median(s_{ij}^{1}, \ldots, s_{ij}^{K}). \tag{4.4}$$

Max

This unsupervised combination method calculates a maximum of K pairwise similarities:

$$s_{ij}^{cmb} = max(s_{ij}^{1}, \ldots, s_{ij}^{K}). \tag{4.5}$$

RankFusion

First, this unsupervised combination method converts each pairwise similarity score s_{ij}^{k} to a rank r_{ij}^{k}. Here, $r_{ij}^{k} = 5$ means that term c_j is the 5-th nearest neighbor of the term c_i, according to the k-th measure. Then, it calculates a combined similarity score as a mean of these ranks:

$$s_{ij}^{cmb} = \frac{1}{K} \sum_{k=1,K} r_{ij}^{k}. \tag{4.6}$$

RelationFusion

This unsupervised combination method keeps only the best relations provided by each measure. All these relations are then merged according to the Algorithm 5. First, the algorithm retrieves the relations extracted by single measures with the k-NN procedure as specified in Section 1.2.2 (line 2):

$$R_k = \bigcup_{i=1}^{|C|} \left\{ \langle c_i, c_j \rangle : (c_j \in top(t, c_i)) \wedge (s_{ij}^k \geq 0) \right\}, s_{ij}^k \in \mathbf{S}_k. \tag{4.7}$$

We have empirically chosen an internal k-NN threshold t of 20%. Then, a set of extracted relations R_k, obtained from the k-th measure, is encoded as an adjacency matrix $\mathbf{R}_k$. An element of this matrix indicates whether terms c_i and c_j are related:

$$r_{ij}^k = \begin{cases} 1 & \text{if semantic relation } \langle c_i, c_j \rangle \in R_k \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

The final similarity score is a mean of adjacency matrices (line 5):

$$\mathbf{S}_{cmb} = \frac{1}{K} \sum_{i=1}^{K} \mathbf{R}_i \Leftrightarrow s_{ij}^{cmb} = \frac{1}{K} \sum_{k=1}^{K} r_{ij}^k. \tag{4.9}$$

Thus, if two measures are combined and the first extracted the relation between c_i and c_j while the second is not, then the similarity s_{ij} will be equal to 0.5.

Algorithm 5: *RelationFusion* combination method.

Input: Similarity matrices produced by N measures, $\{\mathbf{S}_1, \ldots, \mathbf{S}_N\}$
Input: kNN threshold, t
Output: Similarity matrix, $\mathbf{S}_{cmb}$
1 **for** *k=1,N* **do**
2 $R_k \leftarrow threshold(\mathbf{S}_k, t)$;
3 $\mathbf{R}_k \leftarrow relation\_matrix(R_k)$
4 $\mathbf{S}_{cmb} \leftarrow \frac{1}{N} \sum_{k=1}^{N} \mathbf{R}_k$;
5 **return** $\mathbf{S}_{cmb}$;

Logistic Regression

This supervised combination method is based on the Logistic Regression (Hosmer and Stanley, 2000; Agresti, 2002). We train a binary classifier on a set of manually constructed semantic relations R (we use the BLESS and the SN datasets). Positive training examples

are "meaningful" relations (synonyms, hypernyms, etc.), while negative training examples are pairs of semantically unrelated words (generated randomly and verified manually). A semantic relation $\langle c_i, c_j \rangle \in R$ is represented by a vector of pairwise similarities between terms c_i, c_j calculated with K measures $(s_{ij}^1, \ldots, s_{ij}^K)$ and a binary variable r_{ij} (category):

$$r_{ij} = \begin{cases} 0 & \text{if } \langle c_i, c_j \rangle \text{ is a random relation} \\ 1 & \text{otherwise} \end{cases} \tag{4.10}$$

For training and testing of the supervised semantic similarity measures, we use a special 10-fold cross validation ensuring that all relations of one term c are always in the same training/test fold. This modification was needed to calculate correctly the evaluation statistics, such as Precision(10) and Recall(50).

In the Logistic Regression, probability of the i-th relationship is modeled with the Bernoulli distribution:

$$P(r_{ij} = l) = p_i^l \cdot (1 - p_i)^{1-l}, l = \{0, 1\}, \tag{4.11}$$

where the probability of the positive class p_i equals

$$p_i = P(r_{ij} = 1 | s_{ij}^1, \ldots, s_{ij}^K) = \frac{1}{1 + e^{-w_0 + \sum_{k=1}^K w_k s_{ij}^k}}, \tag{4.12}$$

and the probability of the negative class equals

$$1 - p_i = P(r_{ij} = 0 | s_{ij}^1, \ldots, s_{ij}^K) = \frac{e^{-w_0 + \sum_{k=1}^K w_k s_{ij}^k}}{1 + e^{-w_0 + \sum_{k=1}^K w_k s_{ij}^k}}. \tag{4.13}$$

In order to find optimal values of the weights $\mathbf{w}$, the Maximum Likelihood Estimation is used (Agresti, 2002; Theodoridis and Koutroumbas, 2009). We experimented with the Logistic Regression with and without regularization:

- *Logit*. This is a standard unregularized Logistic Regression, which maximizes the following likelihood function:

$$\begin{aligned} L(\mathbf{w}) &= \max_{\mathbf{w}} \sum_{i=1}^N \ln p_i + \sum_{i=1}^N \ln(1 - p_i) \\ &= \max_{\mathbf{w}} \sum_{i=1}^N \ln \frac{1}{1 + e^{-w_0 + \sum_{k=1}^K w_k s_{ij}^k}} + \sum_{i=1}^N \ln \frac{e^{-w_0 + \sum_{k=1}^K w_k s_{ij}^k}}{1 + e^{-w_0 + \sum_{k=1}^K w_k s_{ij}^k}} \end{aligned} \tag{4.14}$$

- *LogitL2*. This version of the Logistic Regression adds a $L2$-regularization term in the likelihood function, e. g. (Fan et al., 2008):

$$L(\mathbf{w}) = \min_{\mathbf{w}} C \sum_{i=1}^N \ln(1 + e^{-w_0 + \sum_{k=1}^K w_k s_{ij}^k}) + \frac{1}{2} \mathbf{w}^T \mathbf{w}. \tag{4.15}$$

- *LogitL1*. This version of the Logistic Regression adds a $L1$-regularization term in the likelihood function, e. g. (Fan et al., 2008):

$$L(\mathbf{w}) = \min_{\mathbf{w}} C \sum_{i=1}^{N} \ln(1 + e^{-w_0 + \sum_{k=1}^{K} w_k s_{ij}^k}) + ||\mathbf{w}||_1. \tag{4.16}$$

These versions of the Logistic Regression rely on some iterative numerical optimization algorithm, such as Newton's method or its variations (Avriel, 2003) to find a vector of weights $\mathbf{w}$ which maximize/minimize the likelihood function $L(\mathbf{w})$. The results of the training are $K + 1$ coefficients of regression $\mathbf{w} = (w_0, w_1, \ldots, w_K)$. We apply the model to combine similarity measures as follows:

$$s_{ij}^{cmb} = \frac{1}{1 + e^{-z}} \text{ where } z = w_0 + \sum_{k=1}^{K} w_k s_{ij}^k. \tag{4.17}$$

Support Vector Machines

This supervised combination method relies on the Support Vector Machines (SVM) (Vapnik, 1999; Cristianini and Shawe-Taylor, 2000; Burges, 1998) trained on the same data as the Logistic Regression described above.

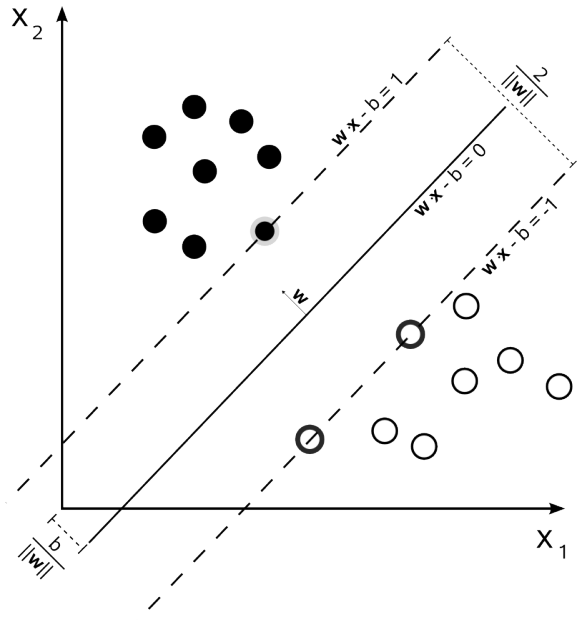


Figure 4.2: A Support Vector Machine: maximal margin hyperplane and its margins.

Let $\mathbf{x}$ be a vector of similarity scores $(s_{ij}^1, \ldots, s_{ij}^K)$ which represent a semantic relation $\langle c_i, c_j \rangle$. Consider the binary classification problem described above. If the positive and the negative training examples can be separated by a hyperplane $\mathbf{w}^T\mathbf{x} - b = 0$, then usually

such separating hyperplane is not unique (see Figure 4.2 [3]). The SVM approaches this problem by selecting a *maximal margin hyperplane*. A *geometrical margin* of such separating hyperplane is the distance to the closest data point:

$$\rho = \frac{\mathbf{w}^T\mathbf{x} - b}{||\mathbf{w}||}. \tag{4.18}$$

These closet points to the separating hyperplane called *support vectors*. To fix the scale of $\mathbf{w}$, one assumes that $|\mathbf{w}^T\mathbf{x} - b| = 1$ if $\mathbf{x}$ is a support vector. Figure 4.2 shows that the support vectors of the positive class lie on the hyperplane: $\mathbf{w}^T\mathbf{x} - b = 1$. The support vectors of the negative class lie on the hyperplane $\mathbf{w}^T\mathbf{x} - b = -1$. An SVM looks for a hyperplane that separates data with the largest margin ρ:

$$\rho = \frac{\mathbf{w}^T\mathbf{x} - b}{||\mathbf{w}||} = \frac{1}{||\mathbf{w}||}. \tag{4.19}$$

The result of a training algorithm is a set of m support vectors $SV = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, where $y_i \in \{+1, -1\}$ is the class label of the vector $\mathbf{x}_i$. The weight vector $\mathbf{w}$ is calculated from these vectors as following:

$$\mathbf{w} = \sum_{x_i \in SV} \alpha_i y_i \mathbf{x}_i. \tag{4.20}$$

Here α_i is a constant obtained in the process of solution of the SVM optimization problem. A reader interested in the details of the optimization procedure should refer to the special literature such as Burges (1998), Vapnik (1999) and Chang and Lin (2011).

We apply the model to combine semantic similarity measures as follows:

$$s_{ij}^{cmb} = \mathbf{w}^T\mathbf{x} + b = \sum_{k=1}^{K} w_i s_{ij}^k + b, \text{ where } K \text{ is the number of features.} \tag{4.21}$$

We experiment with two different versions of the SVM, which formulate their optimization criteria in a slightly different way:

- C-SVM optimizes the following function (Chang and Lin, 2011):

$$\begin{aligned} \min_{\mathbf{w},\xi,b} \quad & \tfrac{1}{2}||\mathbf{w}||^2 + C\textstyle\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(x_i)) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{4.22}$$

  where n is the number of training examples and ξ_i is the variable which measures discrepancy of data point $\mathbf{x}_i$ with the margin. Thus, the constant C lets trade off be-

[3] This image was borrowed from the public domain of the Wikipedia: `http://en.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png`.

tween the margin size and the number of margin errors. Finally, the function $\phi(\mathbf{x}, \mathbf{x}')$ is called *kernel*. It calculates similarity between two feature vectors $\mathbf{x}$ and $\mathbf{x}'$.

- ν-SVM implements mathematically in a slightly different way the intuition about the maximum margin hyperplane. It optimizes the following function (Chang and Lin, 2011):

$$\begin{aligned} \min_{\mathbf{w},\xi,b,\rho} \quad & \tfrac{1}{2}||\mathbf{w}||^2 + \nu\rho + \tfrac{1}{N}\textstyle\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(x_i)) \geq \rho - \xi_i, \\ & \xi_i \geq 0, \rho \geq 0, \end{aligned} \tag{4.23}$$

  where N is the number of training examples. Thus, this kind of SVM introduces an additional meta-parameter $\nu \in [0; 1)$ which has a similar purpose as C of C-SVM. Refer to Schölkopf et al. (2000) for further details.

We are going to experiment with the following kernel functions (refer to Burges (1998) and Cristianini and Shawe-Taylor (2000) for further information):

- *Linear kernel*:

$$\phi(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'. \tag{4.24}$$

- *Polynomial kernel:*

$$\phi(\mathbf{x}, \mathbf{x}') = (\gamma\mathbf{x}^T\mathbf{x}' + c)^b \text{ with } b \in \mathbb{N}, c \geq 0, \gamma > 0. \tag{4.25}$$

- *Gaussian Radial Basis Function (RBF) kernel:*

$$\phi(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2) \text{ with } \gamma = \frac{1}{2\sigma^2}, \sigma \neq 0. \tag{4.26}$$

- *Sigmoid kernel:*

$$\phi(\mathbf{x}, \mathbf{x}') = \tanh(\gamma\mathbf{x}^T\mathbf{x}' + c) \text{ with } \gamma > 0, c \leq 0. \tag{4.27}$$

4.3 Measure Selection Methods

Any of the 9 combination methods presented above may combine from 2 to 16 single measures. Thus, there are

$$\sum_{m=2}^{16} C_{16}^m - 1 = \sum_{m=2}^{16} \frac{16!}{m!(16-m)!} - 1 = 2^{16} - 1 - 16 = 65,519 \tag{4.28}$$

ways to choose which single measures to use in a combination method. We apply three methods to find an efficient combination of measures in this search space: expert choice of measures, forward stepwise procedure and analysis of a Logistic Regression model. Furthermore, the regularized Logistic Regression (*LogitL1* and *LogitL2*) and the Support Vector Machines perform some sort of automatic feature selection.

Expert choice of measures is based on the analytical and empirical properties of the mea-

sures, such as those described in Sections 3.2, 3.3 and 3.4. We chose two sets of respectively 5 and 9 measures which perform well and rely on complementary resources: corpus, Web as a corpus, WordNet, etc. We also selected a group of all measures rejecting only one which has shown the worst results on all datasets. Thus, according to this selection method, we have chosen three groups of measures:

- $E5 = \{3, 9, 10, 13, 14\} = \{$ *Resnik, BDA-3-Cos, SDA-21-Cos, NGD-Factiva, DefVectors-WktWiki* $\}$;
- $E9 = \{1, 3, 9-11, 13-16\} = \{$ *WuPalmer, Resnik, BDA-3-Cos, SDA-21-Cos, PatternSim-Efreq, NGD-Factiva, DefVectors-WktWiki, GlossVectors, ExtendedLesk* $\}$;
- $E15 = \{1, 2, 3, 4, 5, 6, 8-16\} = \{$ *WuPalmer, LeackockChodorow, Resnik, Jiang-Conrath, Lin, NGD-Yahoo, NGD-GoogleWiki, BDA-3-Cos, SDA-21-Cos, PatternSim-Efreq, LSA-Tasa, NGD-Factiva, DefVectors-WktWiki, GlossVectors, ExtendedLesk* $\}$.

Forward stepwise procedure is a greedy algorithm which works as follows. It takes as an input all measures, a combination method and a criterion such as Precision(10). It starts with a void set of measures. Then, at each iteration it adds to the combination one measure which brings the biggest improvement to the criterion. The algorithm stops when no measure can improve the criteria. We used *Mean* as a hybrid measure and the following criteria: Precision(10), Precision(20) and Precision(50). We kept measures which were selected by most of the criteria. According to this method, we have chosen four groups of the measures:

- $S7 = \{9-11, 13-16\} = \{$ *BDA-3-Cos, SDA-21-Cos, PatternSim-Efreq, NGD-Factiva, DefVectors-WktWiki, GlossVectors, ExtendedLesk* $\}$;
- $S8a = \{9-16\} = \{$ *BDA-3-Cos, SDA-21-Cos, PatternSim-Efreq, LSA-Tasa, NGD-Factiva, DefVectors-WktWiki, GlossVectors, ExtendedLesk* $\}$;
- $S8b = \{1, 9-11, 13-16\} = \{$ *WuPalmer, LeackockChodorow, Resnik, JiangConrath, Lin, NGD-Yahoo, NGD-Bing, NGD-GoogleWiki, BDA-3-Cos* $\}$;
- $S10 = \{1, 6, 9-16\} = \{$ *WuPalmer, NGD-Yahoo, BDA-3-Cos, SDA-21-Cos, PatternSim-Efreq, LSA-Tasa, NGD-Factiva, DefVectors-WktWiki, GlossVectors, ExtendedLesk* $\}$.

Automatic feature selection. The last measure selection technique is a method based on analysis of Logistic Regression trained on all 16 measures as features. Only measures with large positive coefficients are selected. According to this method, 12 measures were chosen:

- $R12 = \{3, 5, 6, 8-16\} = \{$ *Resnik, Lin, NGD-Yahoo, NGD-GoogleWiki, BDA-3-Cos, SDA-21-Cos, PatternSim-Efreq, LSA-Tasa, NGD-Factiva, DefVectors-WktWiki, GlossVectors and ExtendedLesk* $\}$.

We test combination methods on the 8 sets of measures specified above. Remarkably, all three selection techniques constantly choose the six following measures – *BDA-3-Cos*, *SDA-21-Cos*, *LSA-Tasa*, *DefVectors-WktWiki*, *GlossVectors*, *ExtendedLesk*.

Similarity Measure	MC	RG	WS	BLESS				SN			
	ρ	*ρ*	*ρ*	*P(10)*	*P (20)*	*P(50)*	*R(50)*	*P(10)*	*P(20)*	*P(50)*	*R(50)*
Random	*0.056*	*-0.047*	*-0.122*	0.546	0.542	0.544	0.522	0.504	0.502	0.499	0.498
1. WuPalmer	0.742	0.775	0.331	0.974	0.929	0.702	0.674	0.982	**0.959**	**0.766**	**0.763**
2. LeackockChodorow	0.724	**0.789**	0.295	0.953	0.901	0.702	0.648	**0.984**	0.953	0.757	0.755
3. Resnik	0.784	0.757	0.331	0.970	0.933	0.700	0.647	0.948	0.908	0.724	0.722
4. JiangConrath	0.719	0.588	0.175	0.956	0.872	0.645	0.458	0.931	0.857	0.625	0.570
5. Lin	0.754	0.619	0.204	0.949	0.884	0.682	0.451	0.939	0.877	0.611	0.566
6. NGD-Yahoo	*0.330*	0.445	0.254	0.940	0.907	0.783	0.648	—	—	—	—
7. NGD-Bing	*0.063*	*0.181*	*0.060*	0.724	0.706	0.650	0.600	0.659	0.619	0.633	0.633
8. NGD-GoogleWiki	*0.334*	0.502	0.251	0.874	0.837	0.703	0.649	—	—	—	—
9. BDA-3-Cos	0.693	0.782	0.466	0.971	0.947	0.836	**0.772**	0.974	0.932	0.742	0.740
10. SDA-21-Cos	**0.790**	0.786	0.491	**0.985**	**0.953**	0.811	0.749	0.978	0.945	0.751	0.743
11. LSA-Tasa	0.694	0.605	0.566	0.968	0.937	0.802	0.740	0.903	0.846	0.641	0.609
12. NGD-Factiva	0.603	0.599	**0.600**	0.959	0.916	0.786	0.681	0.906	0.857	0.731	0.543
13. PatternSim-Efreq	0.461	0.542	0.357	0.972	0.951	**0.944**	0.287	0.920	0.904	0.891	0.295
14. DefVectors-WktWiki	0.759	0.754	0.521	0.943	0.905	0.750	0.679	0.922	0.887	0.725	0.656
15. GlossVectors	0.653	0.738	0.322	0.894	0.860	0.742	0.686	0.932	0.899	0.722	0.709
16. ExtenedLesk	0.792	0.718	0.409	0.937	0.866	0.711	0.657	0.952	0.873	0.655	0.654
Mean-S8a	0.834	0.864	0.734	0.994	0.980	0.870	0.804	0.985	0.965	0.788	0.787
MeanZscore-S8a	0.830	0.864	0.728	0.994	0.981	0.874	0.808	0.986	0.967	0.793	0.792
MeanZscore-S8b	0.844	**0.890**	0.616	0.992	0.977	0.844	0.780	**0.995**	**0.985**	0.815	0.814
MeanNnz-E5	**0.878**	0.878	0.482	0.986	0.956	0.784	0.725	0.975	0.938	0.768	0.766
MeanNnz-S8a	0.843	0.847	**0.740**	0.993	0.977	0.865	0.799	0.986	0.967	0.803	0.802
Median-S10	0.821	0.842	0.647	0.995	0.976	0.843	0.779	0.975	0.934	0.724	0.721
Max-S7	0.802	0.816	0.654	0.979	0.957	0.839	0.775	0.980	0.957	0.786	0.785
RankFusion-S10	—	—	—	0.994	0.978	0.864	0.798	0.976	0.929	0.745	0.744
RelationFusion-S10	—	—	—	**0.996**	0.982	0.840	0.758	0.986	0.963	0.781	0.749
Logit-E15	0.793	0.870	0.690	0.995	**0.987**	**0.885**	**0.818**	**0.995**	0.984	**0.821**	**0.819**

Table 4.1: Performance of single and hybrid similarity measures on the human judgment datasets and the semantic relation ranking task. The best scores in a group are in bold; the best scores in a column are in grey. Correlations in italics mean $p > 0.05$, otherwise $p \leq 0.05$.

4.4 Results

Evaluation of the hybrid measures relies on the tasks of correlation with human judgments about semantic similarity and on the task of semantic relation ranking (see Section 1.2.3). [4] For the first task, as in previous chapters, we use three standard datasets: MC, RG and WordSim. The quality of a measure is assessed with Spearman's correlation with human judgments. For the second task, we use two semantic relation datasets: *BLESS* and *SN*. The quality of a similarity measure is assessed with the four following statistics: Precision(10), Precision(20), Precision(50) and Recall(50).

[4] Results of the experiments described in this section are available at `http://cental.fltr.ucl.ac.be/team/~panchenko/sim-eval/`

4.4.1 General Performance

Table 4.1 and Figure 4.3 present performance of the single and hybrid measures on the five ground truth datasets listed above. The first three columns of the table contain correlations with human judgments, while the other columns present performance on the semantic relation ranking task.

The first part of the table reports on scores of 16 single measures – features used by the combined measures. Our results show that the measures are indeed complementary – no measure performs best on all datasets. For instance, the measure based on a syntactic distributional analysis *SDA-21-Cos* performed best on the MC dataset achieving a correlation of 0.790. The WordNet measure *LeacockChodorow* achieved the top score of 0.789 on the RG dataset. At last, the corpus based measure *NGD-Factiva* was the best on the WordSim dataset, achieving a correlation of 0.600. On the BLESS dataset, syntactic distributional analysis *SDA-21-Cos* performed best for high precision among single measures achieving Precision(20) of 0.953, while the bag-of-words distributional measure *BDA-3-Cos* was the best for high recall with Recall(50) of 0.772. On the SN dataset, the WordNet-based measure *WuPalmer* was the best for both precision and recall.

The second part of Table 4.1 presents performance of the hybrid measures. Our results show that if signals from complementary resources are used, then retrieval of semantically similar words is significantly improved. Most of the hybrid measures outperform the single measures on all the datasets. We tested the 8 first combination methods presented in Section 4.2 with each of the 8 sets of measures specified in Section 4.3. We report the best metrics among all these 64 hybrid measures. The notation *Mean-S8a* means that the *Mean* was used to combine a set of measures *S8a*. We report the pairs, provided the best performance among all possible combinations. As one may observe from Table 4.1, all combination methods perform better on a rich set of 7-15 features, rather than on a well-chosen subset of 2-4 features. The unsupervised combination methods achieve their peak performances with 8-10 features. On the other hand, supervised methods are able to make use of all 15-16 features, improving the performance further.

Measures based on the mean of non-zero similarities (*MeanNnz-S8a* and *MeanNnz-E5*) performed best on MC and WordSim datasets respectively. They achieved correlations of 0.878 and 0.740, which is higher than scores of any other measure. At the same time, measure *MeanZscore-S8b* provided the best scores on the RG dataset among all single and hybrid measures, achieving correlation of 0.890. Supervised measure *Logit-E15* based on Logistic Regression provided the best results on both semantic relation datasets (see Figure 4.3 (a)). Furthermore, it outperformed all single and hybrid measures on that task, in terms of

both precision and recall, achieving Precision(10) of 0.995 and Recall(50) of 0.818 on the BLESS and Precision(10) of 0.995 and Recall(50) of 0.819 on SN.

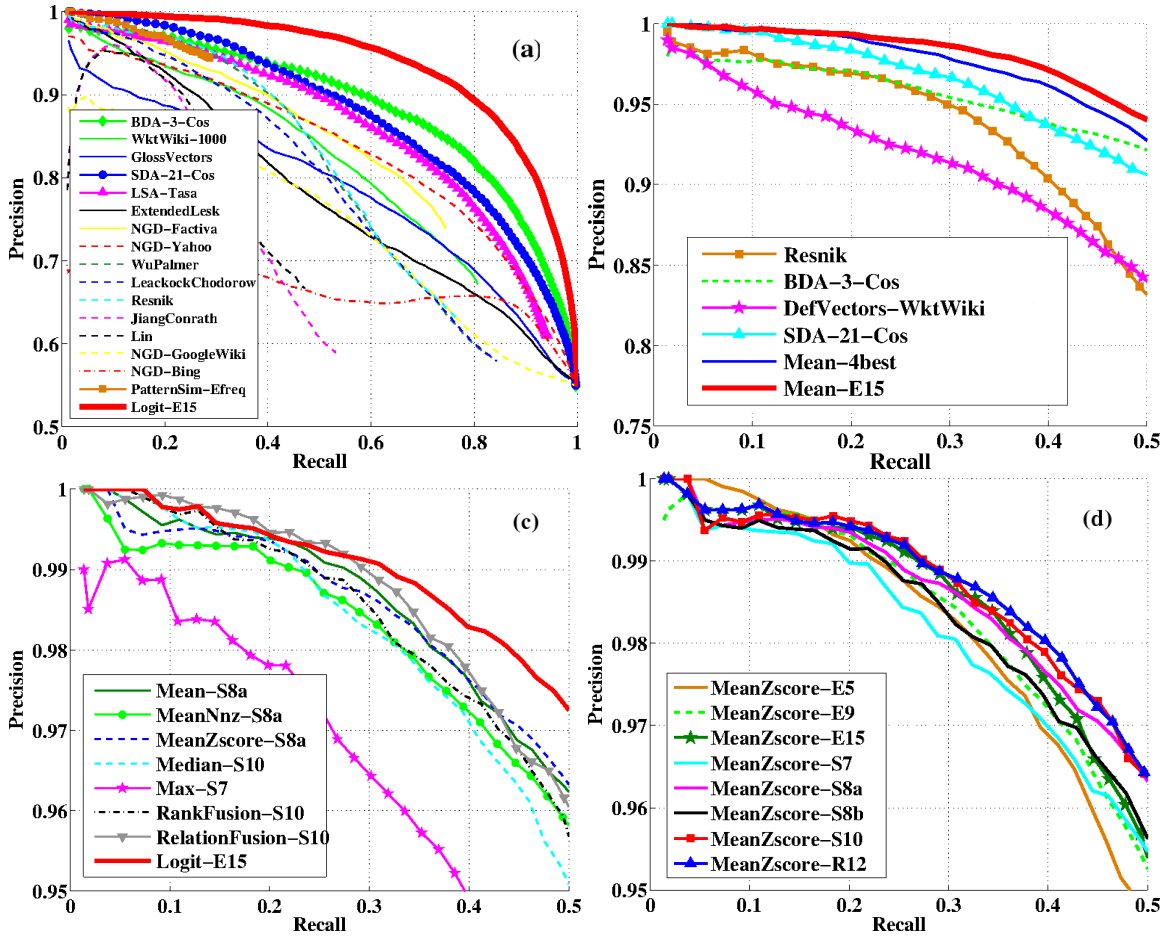


Figure 4.3: Precision-Recall graphs calculated on the BLESS dataset of (a) 16 single similarity measures and the hybrid measure Logit-E15; (b) 4 best single similarity measures, their combination and a combination of 15 measures; (c) 8 hybrid measures; (d) 8 hybrid measures based on the MeanZscore combination method.

Figure 4.3 (b) illustrates another interesting observation. It represents precision-recall curves of the four best performing single measures based on different source of information (*Resnik, BDA-3-Cos, DefVectors-WktWiki* and *SDA-21-Cos*) along with their combination by the *Mean* method. As we can see, a combination of 15 measures performs significantly better than a combination of just 4 measures. Thus, the combination of the four strongest measures can benefit of redundancy provided by the additional weaker measures. However, our results suggest that the difference in performance of the hybrid measures based on *E9, E15, S8a, S8b, S10,* and *R12* is very small (see Figure 4.3 (d)). Thus, if a hybrid measure already combines, for instance, 8 measures, redundancy provided by the additional 4 measures does not change the results drastically.

As we can see in Figure 4.3 (c), combining similarity scores with a *Max* function appears to

be the worst solution. Combination methods based on an average and a median, including *RankFusion* and *RelationFusion*, perform much better. These methods provide quite similar results: in the high precision range, they perform nearly as well as a supervised combination. *RelationFusion* even manages to slightly outperform *Logit* on the first 10-15 k-NN (see Figure 4.3). However, all unsupervised combination methods are significantly worse if higher recall is needed.

We tested some other supervised models to see if they can improve performance of the Logistic Regression. We compared the standard Logistic Regression (*Logit*) with its regularized versions (*LogitL1* and *LogitL2*) and with two kinds of Support Vector Machine (C-SVM and ν-SVM) with various kernels (linear, radial, etc.). The results of this experiment are presented in Table 4.2. The first part of the table lists different models trained on 15 features of the *E15* set (see Section 4.3). Performances of different models, trained on a set of 15 features are similar in most of the cases. The accuracy varies from 0.819 to 0.831 with the exception of C-SVM with the polynomial kernel which achieves accuracy of 0.749. Precision(10) of these models is also comparable and goes from 0.993 to 0.995. In that respect, C-SVMs have a slight performance advantage over ν-SVMs, e. g. accuracy of the *C-SVM-linear-E15* is 0.833, while accuracy of the *ν-SVM-linear-E15* is 0.819. However, the first part of the Table 4.2 compares the models with the default meta-parameters. Tuning these parameters can make the difference between C-SVM and ν-SVM even smaller.

	BLESS					**SN**				
Similarity Measure	*Accu.*	*P(10)*	*P (20)*	*P(50)*	*R(50)*	*Accu.*	*P(10)*	*P(20)*	*P(50)*	*R(50)*
C-SVM-linear-E15	**0.833**	0.995	**0.986**	**0.884**	**0.817**	0.820	0.995	0.981	0.816	0.816
C-SVM-poly-E15	0.749	0.993	0.976	0.798	0.737	0.795	0.993	0.977	0.791	0.791
C-SVM-radial-E15	0.832	**0.996**	**0.986**	0.883	0.816	**0.831**	0.995	**0.988**	**0.838**	**0.839**
C-SVM-sigmoid-E15	0.829	0.995	0.985	0.881	0.813	0.811	0.995	0.986	0.807	0.808
ν-SVM-radial-E15	0.827	0.995	0.985	0.879	0.812	0.815	**0.996**	0.984	0.811	0.811
ν-SVM-linear-E15	0.819	**0.996**	0.984	0.877	0.810	0.805	0.994	0.984	0.803	0.803
ν-SVM-poly-E15	0.827	**0.996**	0.985	0.879	0.812	0.826	0.995	**0.988**	0.833	0.833
ν-SVM-sigmoid-E15	0.827	0.995	0.984	0.878	0.811	0.811	0.995	0.984	0.809	0.809
Logit-E15	0.831	0.994	**0.986**	**0.884**	**0.817**	0.823	0.994	0.983	0.819	0.819
LogitL2-E15	0.823	0.995	0.982	0.874	0.808	0.773	0.990	0.967	0.798	0.798
LogitL1-E15	0.824	0.994	0.984	0.874	0.807	0.787	0.992	0.975	0.805	0.805
Logit-E5	0.796	0.989	0.977	0.853	0.788	0.795	0.985	0.965	0.791	0.791
C-SVM-radial-E5	0.802	0.990	0.976	0.857	0.792	0.788	0.980	0.959	0.787	0.787
Logit-E9	0.821	0.991	0.983	0.877	0.810	0.821	0.995	0.982	0.824	0.824
C-SVM-radial-E9	0.824	0.993	0.983	0.875	0.809	0.831	**0.997**	0.988	0.837	0.837
Logit-E15	**0.831**	**0.995**	**0.986**	**0.884**	**0.817**	**0.832**	0.995	**0.989**	**0.840**	**0.839**
C-SVM-radial-E15	**0.831**	0.994	**0.986**	**0.884**	**0.817**	0.823	0.994	0.983	0.819	0.819
C-SVM-radial-E15 ($C = 32, \gamma = 2$)	**0.855**	0.987	0.979	**0.900**	**0.831**	**0.846**	0.983	0.981	**0.846**	**0.846**
C-SVM-radial-E15 ($C = 32, \gamma = .125$)	0.841	**0.996**	**0.987**	0.892	0.824	0.844	**0.995**	**0.990**	0.845	0.845

Table 4.2: Performance of the hybrid supervised semantic similarity measures.

According to our results, the following three combination methods have a slight advantage over the other supervised methods: *C-SVM-linear*, *C-SVM-radial* and *Logit*. Furthermore, the Support Vector Machine with the Gaussian Radial Basis Function kernel (*C-SVM-radial*)

outperforms all others in terms of Precision(10) and Precision(20) on the BLESS dataset and in terms of Accuracy, Precision(20), Precision(50) and Recall(50) on the SN dataset. However, this difference is not statistically significant in most of the cases.

The second part of the Table 4.2 compares performance of the Logistic Regression (*Logit*) and the SVM (*C-SVM-radial*) trained on different feature sets (*E5*, *E9*, *E15*) described in Section 4.3. One may observe that all performance statistics grow as the number of features grow. On the other hand, performance of the Logistic Regression and the SVM trained on the same features are very similar and their difference is not statistically significant.

The results presented in the first two parts of Table 4.2 were obtained with default meta-parameters of the models (see Section 4.2, Fan et al. (2008) and Chang and Lin (2011)):

- Logistic Regression (*Logit*): no;
- L1-regularized Logistic Regression (*LogitL1*): cost $C = 1$, no bias term;
- L2-regularized Logistic Regression (*LogitL2*): cost $C = 1$, no bias term;
- C-SVMs: cost $C = 1$;
- ν-SVMs: $\nu = 0.5$;
- Linear kernel: no;
- RBF kernel: cost $C = 1$, $\gamma = \frac{1}{\#features}$;
- Polynomial kernel: cost $C = 1$, $\gamma = \frac{1}{\#features}$, degree $b = 3$, free coefficient in the kernel function $c = 0$;
- Sigmoid kernel: cost $C = 1$, $\gamma = \frac{1}{\#features}$, free coefficient of the kernel $c = 0$.

Figure 4.4 presents results of the meta-parameter optimization of the C-SVM with RBF kernel (*C-SVM-radial-E15*). Each plot is a two dimensional contour graph with the contour interval of 1%. The third part of the Table 4.2 provides performance scores of the two most prominent parameter configurations found by this grid search. As one can observe, the models have relatively small variance and a sub-optimal choice of the mataparameters does not ruin the performance. In our case, the models with large values of C achive better results on the both benchmarks. Furthermore, the combination of $C = 32$ and $\gamma = 2$ yielded the best results in terms of Accuracy, Precision(50) and Recall(50) both for the BLESS and SN datsets. However, to maximize the Precision(10) one should select a slightly different configuration of the metaparameters: $C = 32$ and $\gamma = 0.125$. Nonetheless, the same level of Precision(10) is achieved with the default combination of the metaparameters (see Table 4.2). Thus, the meta-parameter optimization improves performance by 1-3% depending on the statistic. It may be especially useful if we need to maximize a particular statistic, such as Precision(10) or Recall(50).

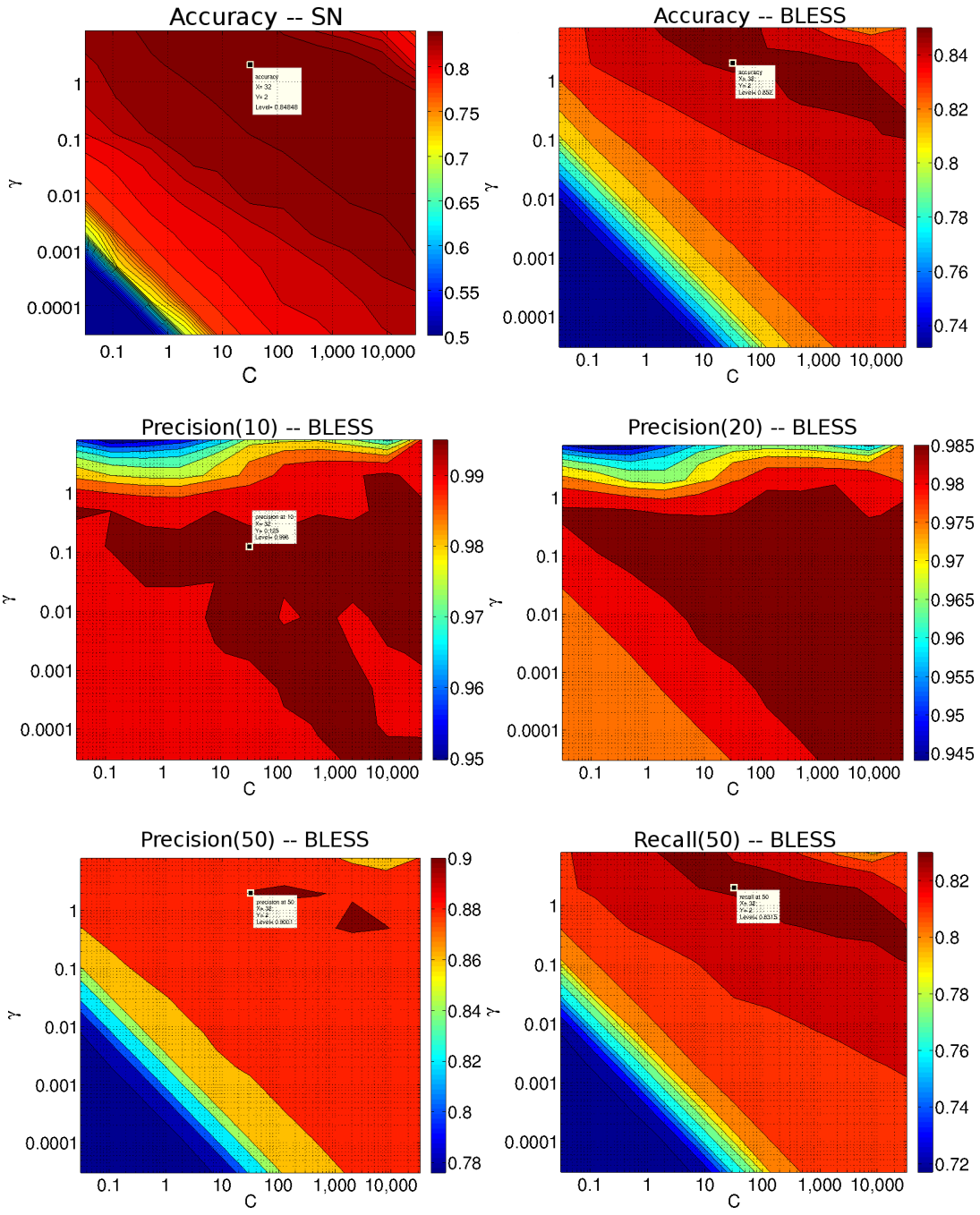


Figure 4.4: Meta-parameter optimization with the grid search of the C-SVM-radial-E15 measure: Accuracy, Precision(10), Precision(20), Precision(50) and Recall(50).

We conclude that the supervised combination methods (e. g., *Logit-E15*, *C-SVM-radial-E15* or *C-SVM-linear-E15*) outperform all single and hybrid unsupervised measures on all datasets examined in this chapter. On the other hand, advantage of one supervised combination method over the others is less clear. In the following section we further examine one of the most successful supervised models found in this chapter – *Logit-E15*.

4.4.2 Semantic Relation Distribution of the Hybrid Measure *Logit-E15*

Figure 4.5 presents an analysis of lexico-semantic knowledge captured by the *Logit-E15* measure. Figure 4.5 (a) presents the distributions of similarities across various relation types. These distributions were calculated based on the BLESS dataset as suggested in (Baroni and Lenci, 2011). This analysis of semantic relation distribution is similar to the one presented in Section 3.6.3 for the single measures. First, scores of the meaningful relation types (hypernyms, co-hypernyms, meronyms, events and attributes) are significantly higher than scores of the random pairs. This is clear from Figure 4.5 (b), which represents results of the ANOVA on these scores. This means that we can systematically separate meaningful word-pairs from the random ones. We can also observe from this plot that there is no significant difference in the scores assigned to pairs of hypernyms, co-hyponyms, meronyms and words related by the event relation. On the other hand, as one may see, pairs of words related with attribute relation are scored significantly lower. This result may be interpreted as a positive one as we are looking for a measure which score high hypernyms, co-hyponyms and synonyms.

Figure 4.5 (c) illustrates yet another property of the semantic relation distribution of the *Logit-E15*. This figure plots distributions of similarity scores of several single measures used by the *Logit-E15*. As we mentioned above, the hybrid measure assigns high scores to hypernyms, co-hyponyms, meronyms and event relations (see Figure 4.5). On the other hand, single measures score differently relations of these types. For instance, *PatternSim-Efreq* assigns the highest scores to hypernyms and co-hyponyms and very low scores to all other types. *LSA-Tasa* scores high pairs of co-hyponyms, meronyms and event relations. *SDA-21-Cos* assigns the very high scores to co-hyponyms, high scores to hypernyms and low scores to relations of other types. Hence, the fusion of different measures smooths the preference for a specific relation type.

Finally, Table 4.3 presents a toplist of word-pairs from the BLESS and the SN datasets sorted by the similarity score computed by the *Logit-E15* measure. Each target word has only few related words with the highest similarity score. One can observe that co-hyponyms are prevalent among the top-ranked relations.

4.5 Discussion

The hybrid measures achieve higher precision and recall than the single measures. First, it is due to the common lexico-semantic information, such as that a “car” is a synonym of a “vehicle”, provided by the knowledge- and definition-based measures. Measures based on WordNet and dictionary definitions achieve high precision as they rely on fine-grained man-

BLESS Dataset				SN Dataset			
Target, c_i	**Type**	**Relatum,** c_j	**Score,** s_{ij}	**Target,** c_i	**Type**	**Relatum,** c_j	**Score,** s_{ij}
acacia	cohypo	birch	1	abuse	cohypo	contumely	1
acacia	cohypo	cypress	1	abuse	cohypo	ill-treat	1
acacia	cohypo	elm	1	abuse	cohypo	ill-treatment	1
acacia	cohypo	oak	1	abuse	cohypo	insult	1
acacia	cohypo	pine	1	abuse	cohypo	insults	1
acacia	hyper	tree	1	abuse	cohypo	maltreat	1
acacia	cohypo	willow	1	abuse	cohypo	maltreatment	1
alligator	cohypo	crocodile	1	abuse	cohypo	mistreat	1
alligator	cohypo	lizard	1	abuse	cohypo	misuse	1
alligator	cohypo	snake	1	abuse	cohypo	rape	1
alligator	cohypo	turtle	1	advocate	cohypo	proponent	1
ambulance	mero	paramedic	1	agriculture	cohypo	agronomy	1
ant	cohypo	bee	1	agriculture	cohypo	cultivation	1
ant	cohypo	beetle	1	agriculture	cohypo	farming	1
ant	cohypo	butterfly	1	agriculture	cohypo	floriculture	1
ant	cohypo	cockroach	1	agriculture	cohypo	horticulture	1
ant	cohypo	dragonfly	1	agriculture	cohypo	husbandry	1
ant	cohypo	grasshopper	1	ague	cohypo	exanthem	1
ant	cohypo	hornet	1	ague	cohypo	mumps	1
ant	hyper	insect	1	ague	cohypo	polio	1
ant	cohypo	mosquito	1	airplane	cohypo	aeroplane	1
ant	cohypo	moth	1	airplane	cohypo	aircraft	1
ant	cohypo	silverfish	1	airplane	cohypo	airliner	1
ant	cohypo	wasp	1	airplane	cohypo	jet	1
apple	cohypo	apricot	1	airplane	cohypo	plane	1
apple	cohypo	banana	1	airship	cohypo	blimp	1
apple	cohypo	cherry	1	airship	cohypo	dirigible	1
apple	cohypo	cranberry	1	airship	cohypo	zeppelin	1
apple	hyper	fruit	1	alcohol	cohypo	alcoholism	1
apple	cohypo	grape	1	alcohol	cohypo	ethanol	1
apple	cohypo	grapefruit	1	alcohol	cohypo	intoxicant	1
apple	mero	juice	1	alcohol	cohypo	methanol	1
apple	cohypo	lemon	1	alligator	cohypo	crocodile	1
apple	cohypo	lime	1	alligator	cohypo	gator	1
apple	cohypo	peach	1	alloy	cohypo	metal	1
apple	cohypo	pear	1	anarchist	cohypo	nihilist	1
apple	cohypo	pineapple	1	anarchist	cohypo	syndicalist	1
apple	cohypo	plum	1	animal	cohypo	beast	1
apple	cohypo	strawberry	1	animal	cohypo	creature	1
apricot	cohypo	apple	1	animal	cohypo	fauna	1
apricot	cohypo	banana	1	architect	cohypo	bricklayer	1
apricot	cohypo	cherry	1	architect	cohypo	builder	1
apricot	hyper	fruit	1	architect	cohypo	designer	1
apricot	cohypo	grape	1	architect	cohypo	engineer	1
apricot	cohypo	grapefruit	1	aristocrat	cohypo	noble	1
apricot	mero	juice	1	aristocrat	cohypo	patrician	1
apricot	cohypo	lemon	1	arm	cohypo	arms	1
apricot	attri	orange	1	arm	cohypo	limb	1
apricot	cohypo	peach	1	armor	cohypo	armour	1
apricot	cohypo	peach	1	armor	cohypo	armour	1
apricot	cohypo	pear	1	armor	cohypo	breastplate	1
apricot	cohypo	pineapple	1	armor	cohypo	cuirass	1
apricot	cohypo	plum	1	armor	cohypo	helmet	1
apricot	cohypo	strawberry	1	armor	cohypo	shield	1
axe	mero	blade	1	arm	cohypo	weapon	1
axe	cohypo	chisel	1	army	cohypo	soldiers	1
axe	event	chop	1	art	cohypo	artistry	1
axe	cohypo	dagger	1	artifact	cohypo	artefact	1
axe	cohypo	hammer	1	artist	cohypo	designer	1
axe	cohypo	hatchet	1	artist	cohypo	musician	1
axe	cohypo	knife	1	artist	cohypo	painter	1
axe	cohypo	scissors	1	art	cohypo	painting	1
axe	cohypo	shovel	1	art	cohypo	prowess	1
axe	cohypo	spear	1	aspirant	cohypo	hopeful	1
axe	cohypo	sword	1	atheist	cohypo	agnostic	1
bag	cohypo	backpack	1	atheist	cohypo	deist	1
bag	cohypo	basket	1	atheist	cohypo	freethinker	1
bag	cohypo	bottle	1	atheist	cohypo	rationalist	1
bag	cohypo	box	1	atheist	cohypo	skeptic	1
bag	hyper	container	1	athletics	cohypo	gymnastics	1
bag	hyper	luggage	1	attack	cohypo	assail	1
bag	mero	plastic	1	attack	cohypo	assault	1
bag	cohypo	pouch	1	attack	cohypo	attempt	1
bag	cohypo	pouch	1	attack	cohypo	onset	1
bag	cohypo	purse	1	attack	cohypo	onslaught	1
bag	cohypo	suitcase	1	attack	cohypo	raid	1
bag	cohypo	wallet	1	audience	cohypo	listeners	1
bag	cohypo	wallet	1	audience	cohypo	spectators	1
banana	cohypo	apple	1	audience	cohypo	viewers	1
banana	cohypo	apricot	1	authoritarianism	cohypo	despotism	1
banana	cohypo	cherry	1	authoritarianism	cohypo	dictatorship	1
banana	cohypo	coconut	1	authoritarianism	cohypo	fascism	1
banana	hyper	fruit	1	authoritarianism	cohypo	totalitarianism	1
banana	cohypo	lemon	1	authoritarianism	cohypo	tyranny	1
banana	cohypo	mango	1	authority	cohypo	agency	1
banana	cohypo	papaya	1	authority	cohypo	authorization	1
banana	cohypo	peach	1	authority	cohypo	dominance	1
banana	cohypo	pineapple	1	authority	cohypo	jurisdiction	1
banana	cohypo	plum	1	authority	cohypo	power	1
banana	cohypo	strawberry	1	authority	cohypo	sanction	1
banana	attri	sweet	1	autocrat	cohypo	despot	1
battleship	cohypo	frigate	1	autocrat	cohypo	tyrant	1
battleship	hyper	warship	1	automobile	cohypo	auto	1
bear	cohypo	bull	1	automobile	cohypo	car	1

Table 4.3: The hybrid similarity measure Logit-E15: toplist of word-pairs sorted by similarity score from the BLESS dataset (on the left) and the SN dataset (on the right).

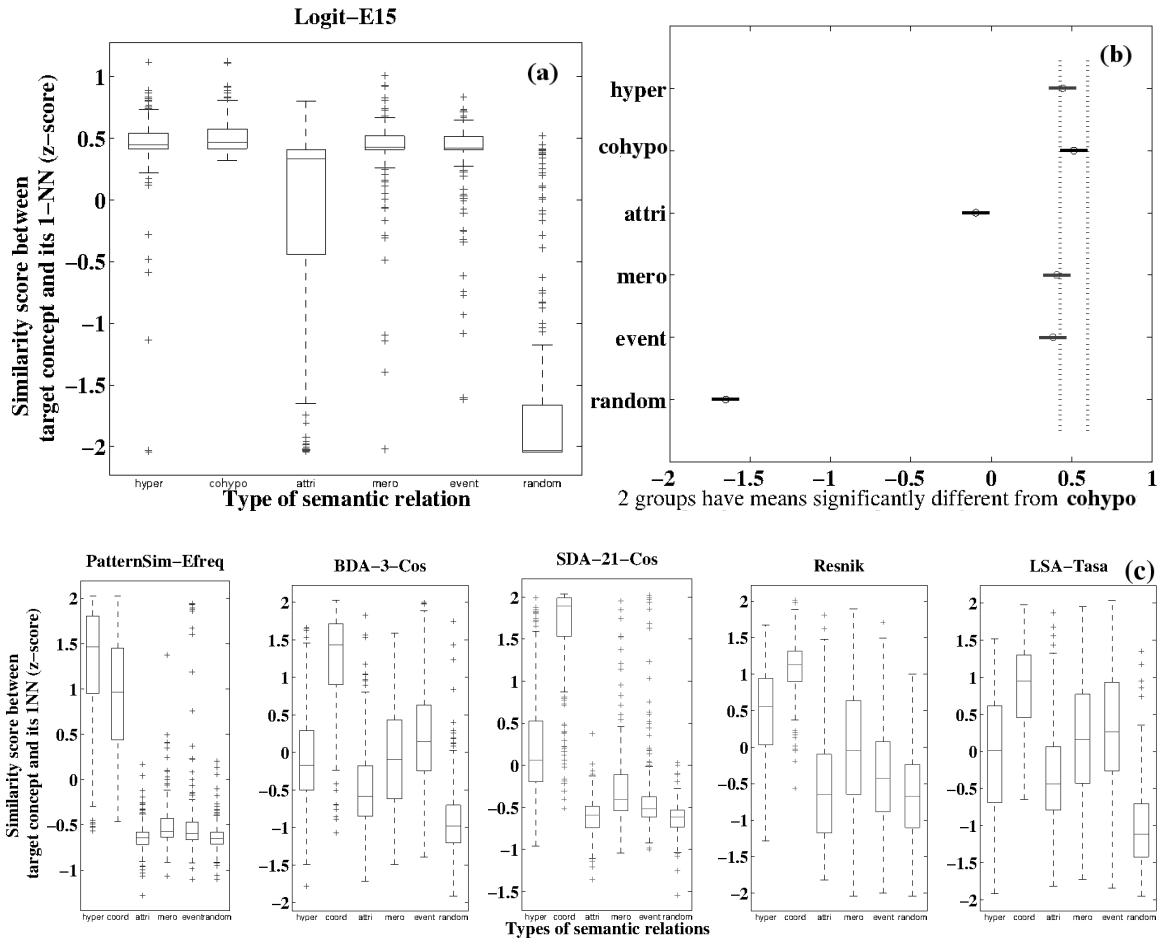


Figure 4.5: (a) Distribution of 1-NN similarity scores of the Logit-E15 similarity measure on the BLESS dataset; (b) ANOVA analysis of the 1-NN similarity scores; (c) distribution of 1-NN similarity scores of a five single measures (PatternSim-Efreq, BDA-3-Cos, SDA-21-Cos, Resnik and LSA-Tasa) used as features by the hybrid measure Logit-E15.

ually constructed resources. However, due to limited coverage of these resources, they only can determine relations between a limited number of terms. On the other hand, measures based on web and corpora are nearly unlimited in their coverage, but provide less precise results. The combination of the measures enables keeping high precision for frequent terms (e. g., "disease") present in WordNet and dictionaries and empowers calculation of relations between rare terms unlisted in the handcrafted resources (e. g., "bronchocele") with web and corpus measures.

Second, combinations work well because, as it was found in previous research (Sahlgren, 2006; Heylen et al., 2008) and demonstrated in the previous chapter, different measures provide complementary types of semantic relations. For instance, WordNet-based measures score higher hypernyms than associative relations, distributional analysis score high co-hyponyms and synonyms, etc. In that respect, a combination helps to recall more different relations. For example, a WordNet-based measure may return a hypernym ⟨salmon, seafood⟩, while a corpus-based measure would extract a co-hyponym ⟨salmon, mackerel⟩.

Figure 4.6 illustrates these benefits of the hybrid measure on example of the *Logit-E15* measure. It plots similarity scores between 74 words related to the word "acacia" in the BLESS dataset. In this plot, the word-pairs are sorted according to relation types:

- 1 is a hypernym;
- 2-10 are co-hyponyms;
- 11-24 are meronyms;
- 25-33 are event relations;
- 34-41 are attribute relations;
- 42-74 are random relations.

Thus, the positions 1-41 represent meaningful relations, while the positions 42-74 represent random relations (see Table 4.5). The Y-axis represents a similarity score s_{ij} between a corresponding pairs of words $\langle c_i, t, c_j \rangle$. First, one may notice that the single measures are complementary in their coverage. Second, the combined model makes decisions more robustly. For instance, *Logit-E15* assigned the maximum score to all pairs of hypernyms and co-hyponyms as several single measures assigned high scores to these pairs. Notice that not all single models assigned high scores to all hypernyms and co-hypernyms. However, the combined model assigned the maximum score to all the hypernyms and co-hypernyms. Thus, the combined measure smooths scores of true positives (see the last plot of Figure 4.6). The three clear false positives correspond to the pairs $\langle acacia, random, open \rangle$, $\langle acacia, random, fading \rangle$ and $\langle acacia, random, cover \rangle$. The four most distinct false negatives correspond to the following pairs:

- $\langle acacia, event, live \rangle$,
- $\langle acacia, event, die \rangle$,
- $\langle acacia, attri, odorous \rangle$,
- $\langle acacia, event, cut \rangle$.

Finally, the supervised combination method *Logit* works better than the unsupervised ones (*Mean, MeanZscore* and others) because of two reasons. First, the measures generate scores which have quite different distributions on the range $[0; 1]$ (see Table 4.4). Averaging of such scores may be suboptimal. Logistic Regression overcomes this issue by assigning appropriate weights $(w_1, \ldots, w_K)$ to the measures in the linear combination:

$$z = w_0 + \sum_{k=1}^{K} w_k \cdot s_{ij}^k. \tag{4.29}$$

Second, training procedure enables the model to assign higher weights to the measures

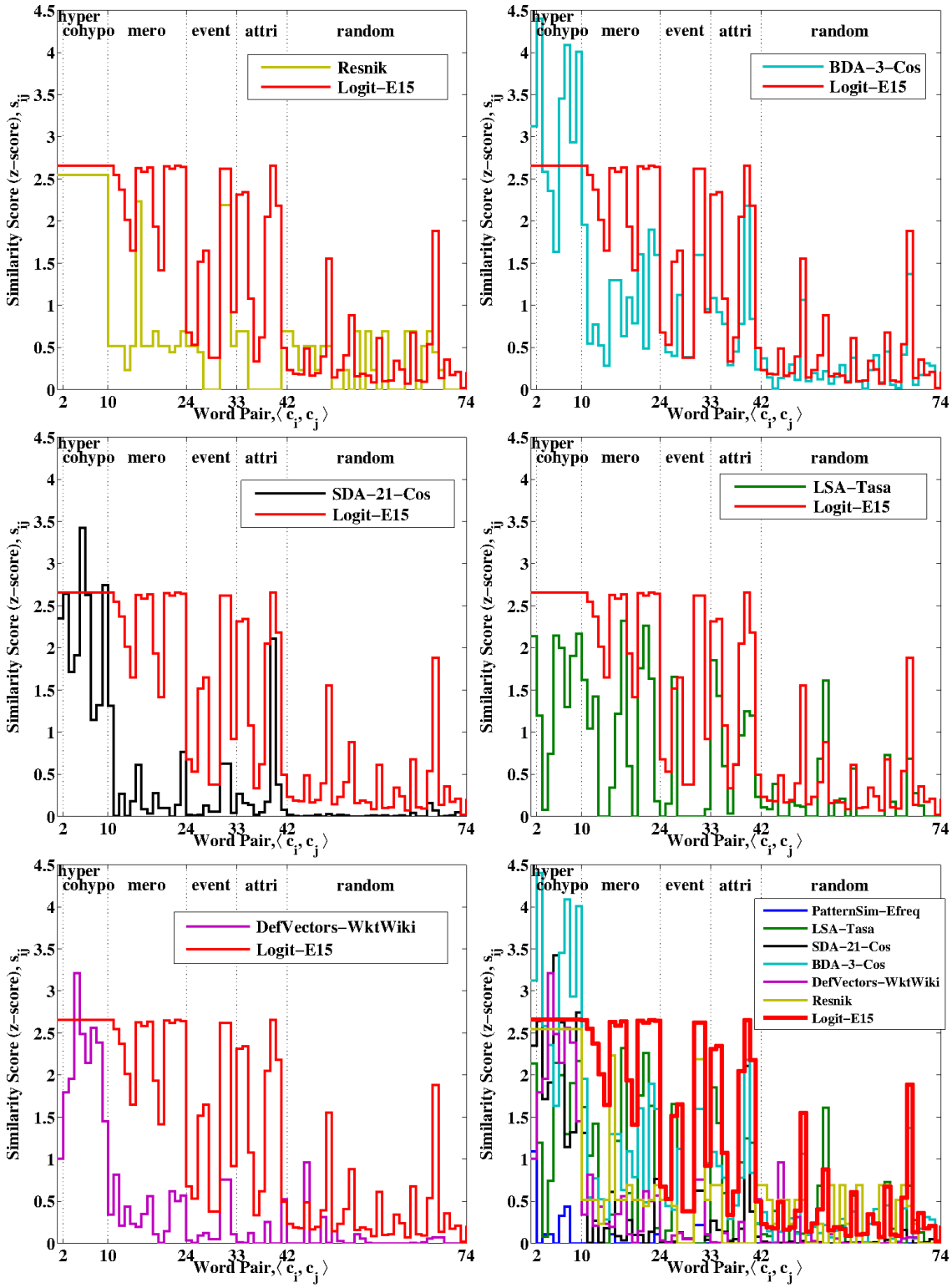


Figure 4.6: Similarity scores between 74 words related to word "acacia" in the BLESS dataset (see Table 4.5). The scores were calculated by PatternSim-Efreq, BDA-3-Cos, SDA-21-Cos, Resnik, LSA-Tasa, DefVectors-WktWiki and Logit-E15. The original scores were transformed into z-scores.

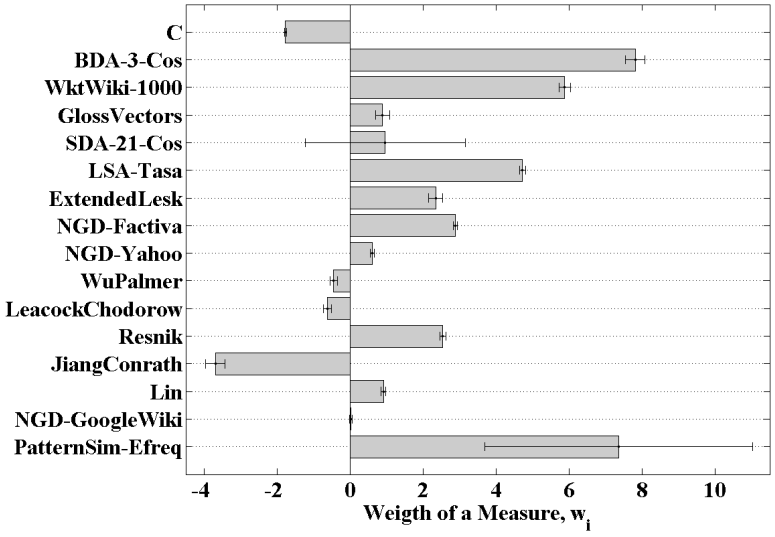


Figure 4.7: Weights of the similarity measures used by the hybrid measure Logit-E15. The weights were learnt on the BLESS dataset with 10-fold cross validation repeated 10 times.

which provide better results, while an averaging process sets equal weights. Figure 4.7 depicts a distribution of weights $(w_0, w_1, \ldots, w_{15})$ of the similarity measure *Logit-E15*. Here w_0 is a free coefficient and other coefficients correspond to the 15 measures from the *E15* set specified in Section 4.3 (see Table 4.4). The figure summarizes weights of some 100 models learnt on the BLESS dataset (10 runs of a 10-fold cross-validation).

Weight	**Measure**	$median(s_{ij})$	$\mu(s_{ij})$	$\sigma(s_{ij})$
w_1	BDA-3-Cos	0.0528	0.0972	0.1314
w_2	DefVectors-WktWiki	0.0052	0.0546	0.1283
w_3	GlossVectors	0.0154	0.0347	0.0656
w_4	SDA-21-Cos	0.0020	0.0210	0.0669
w_5	LSA-Tasa	0.0859	0.1297	0.1413
w_6	ExtendedLesk	0.0320	0.0565	0.0950
w_7	NGD-Factiva	0.0309	0.1128	0.1575
w_8	NGD-Yahoo	0.5486	0.5378	0.1662
w_9	WuPalmer	0.4615	0.4330	0.2963
w_{10}	LeacockChodorow	0.3500	0.3216	0.2067
w_{11}	Resnik	0.1164	0.1994	0.2250
w_{12}	JiangConrath	0	0.0169	0.0502
w_{13}	Lin	0	0.1219	0.2159
w_{14}	NGD-GoogleWiki	0.6222	0.6055	0.1400
w_{15}	PatternSim-Efreq	0	0.0052	0.0296

Table 4.4: Similarity scores of the single measures (features) used by the hybrid measure Logit-E15.

The advantages of the hybrid measures come at the cost of higher computational complexity. To compute a similarity score with a hybrid measure sim_{cmb}, we need to calculate the scores of all single measures $\{sim_i\}_i$ used in a combination and then apply a combination method

cmb:

$$O(sim_{cmb}) = \sum_i O(sim_i) + O(cmb) = \sum_i O(sim_i) + O(1). \quad (4.30)$$

Computational complexity of the combination methods used in this chapter (see Section 4.2) are constant: $O(cmb) = O(1)$. For the Logistic Regression, this corresponds to the time needed to apply a linear combination of similarity scores: $s_{ij}^{cmb} = (1 + \exp(-\sum_k^K w_k s_{ij}^k + w_0))^{-1}$.

Thus, the complexity of the *Logit-E15* measure is equal to

$$O(\textit{Logit-E15}) = \sum_{sim_i \in E15} O(sim_i) \text{ where } E15 = \{\textit{WuPalmer, LeackockChodorow, Resnik, JiangConrath, Lin, NGD-Yahoo, NGD-GoogleWiki, BDA-3-Cos, SDA-21-Cos PatternSim-Efreq, LSA-Tasa, NGD-Factiva, DefVectors-WktWiki, GlossVectors, ExtendedLesk} + O(1). \quad (4.31)$$

The space complexity of the hybrid similarity score is calculated in the same way as the computational complexity. In this case, the computational complexities of the single measures will be replaced by the corresponding space complexities.

4.6 Conclusion

In this chapter, we presented several hybrid semantic similarity measures based on the 16 single measures combined with 9 fusion methods and 3 feature selection techniques. The combined measures were evaluated on the correlations with human judgments and on the semantic relation ranking task (see Section 1.2.3). Our results have shown that the hybrid measures outperform the state-of-the-art single measures on all these benchmarks. In particular, the techniques that combine 15 corpus-, web-, network-, and dictionary-based measures with the supervised models provided the best results (*Logit-E15*, *C-SVM-linear-E15*, *C-SVM-radial-E15*). For instance, the *Logit-E15* measure achieves a correlation with human judgments of 0.870, Precision(10) of 0.995 and Recall(50) of 0.818. Our experiments have shown that the supervised combination methods are able to make use of several highly correlated variables. These measures rely on the "robustness via redundancy" principle: a combination of the strongest predictors is strengthened by the redundancy provided by the additional "weaker" predictors.

N	**Target, c_i**	**Type**	**Relatum, c_j**	**PatternSim-Efreq**	**LSA-Tasa**	**BDA-3-Cos**	**SDA-21-Cos**	**DefVectors-WktWiki**	**Resnik**	**Logit-E15**
1	acacia	hyper	tree	1.09	2.14	2.35	3.13	1.00	2.55	2.66
2	acacia	cohypo	birch	0.00	1.20	2.64	4.41	1.79	2.55	2.66
3	acacia	cohypo	cedar	0.11	0.08	1.71	2.58	1.95	2.55	2.66
4	acacia	cohypo	cypress	0.11	0.74	1.91	2.36	3.21	2.55	2.66
5	acacia	cohypo	elm	0.00	2.15	3.42	1.63	2.48	2.55	2.66
6	acacia	cohypo	oak	0.33	2.00	2.62	3.45	2.15	2.55	2.66
7	acacia	cohypo	pine	0.44	1.30	1.14	4.09	2.56	2.55	2.66
8	acacia	cohypo	poplar	0.00	1.90	1.32	2.93	2.39	2.55	2.66
9	acacia	cohypo	willow	0.00	2.17	2.74	4.01	1.45	2.55	2.66
10	acacia	mero	bark	0.00	1.62	1.31	1.96	0.34	0.52	2.66
11	acacia	mero	bole	0.00	1.04	0.00	0.54	0.82	0.52	2.55
12	acacia	mero	branch	0.00	1.42	0.26	0.77	0.21	0.52	2.37
13	acacia	mero	burl	0.00	0.00	0.01	0.52	0.43	0.23	2.01
14	acacia	mero	crown	0.00	0.00	0.18	0.28	0.23	0.52	1.65
15	acacia	mero	flower	0.11	0.34	0.61	1.30	0.20	2.23	2.63
16	acacia	mero	leaf	0.00	1.21	0.09	1.30	0.35	0.52	2.59
17	acacia	mero	limb	0.00	2.32	0.03	0.63	0.56	0.52	2.63
18	acacia	mero	root	0.00	0.59	0.28	1.10	0.11	0.69	1.93
19	acacia	mero	spike	0.00	0.00	0.10	0.78	0.01	0.52	1.41
20	acacia	mero	stump	0.00	1.76	0.10	1.61	0.15	0.52	2.65
21	acacia	mero	treetop	0.00	2.26	0.00	0.49	0.62	0.44	2.62
22	acacia	mero	trunk	0.00	1.64	0.14	1.90	0.50	0.52	2.65
23	acacia	mero	wood	0.11	0.18	0.76	1.60	0.57	0.69	2.64
24	**acacia**	**event**	**cut**	0.00	0.00	0.02	0.68	0.03	0.52	**0.67**
25	**acacia**	**event**	**die**	0.00	0.15	0.01	0.44	0.00	0.52	**0.53**
26	acacia	event	fall	0.00	1.66	0.02	0.40	0.09	0.44	1.52
27	acacia	event	grow	0.00	0.00	0.13	1.13	0.12	0.00	1.65
28	**acacia**	**event**	**live**	0.00	0.00	0.05	0.38	0.05	0.00	**0.38**
29	**acacia**	**event**	**live**	0.00	0.00	0.05	0.38	0.05	0.00	**0.38**
30	acacia	event	plant	0.22	0.00	0.62	1.60	0.76	2.19	2.62
31	acacia	event	plant	0.22	0.00	0.62	1.60	0.76	2.19	2.62
32	acacia	event	stand	0.00	0.08	0.04	0.95	0.10	0.52	0.92
33	acacia	attri	brown	0.00	1.85	0.27	1.08	0.01	0.69	2.31
34	acacia	attri	green	0.00	1.43	0.16	0.92	0.01	0.69	2.34
35	acacia	attri	large	0.00	0.59	0.14	0.78	0.12	0.00	1.08
36	**acacia**	**attri**	**odorous**	0.00	0.04	0.02	0.29	0.00	0.00	**0.34**
37	acacia	attri	old	0.00	0.61	0.05	0.45	0.00	0.00	0.62
38	acacia	attri	tall	0.00	0.96	0.17	0.78	0.25	0.00	2.05
39	acacia	attri	thorny	0.00	1.25	2.11	2.18	0.00	0.00	2.66
40	acacia	attri	yellow	0.00	1.19	0.38	0.84	0.00	0.00	2.18
41	acacia	attri	young	0.00	0.23	0.08	0.24	0.53	0.69	0.50
42	acacia	random	begin	0.00	0.10	0.02	0.37	0.00	0.69	0.23
43	acacia	random	boat	0.00	0.08	0.01	0.14	0.00	0.52	0.19
44	acacia	random	conductivity	0.00	0.39	0.00	0.01	0.00	0.23	0.18
45	acacia	random	connexion	0.00	0.00	0.00	0.14	0.96	0.52	0.49
46	acacia	random	consolidation	0.00	0.00	0.01	0.15	0.00	0.52	0.16
47	acacia	random	content	0.00	0.17	0.02	0.29	0.00	0.23	0.19
48	acacia	random	cortex	0.00	0.13	0.00	0.11	0.31	0.52	0.39
49	**acacia**	**random**	**cover**	0.00	0.11	0.03	1.07	0.01	0.52	**1.55**
50	acacia	random	democracy	0.00	0.00	0.01	0.10	0.14	0.00	0.14
51	acacia	random	disease	0.00	0.00	0.02	0.19	0.03	0.00	0.23
52	acacia	random	eyelid	0.00	0.68	0.01	0.12	0.00	0.23	0.40
53	**acacia**	**random**	**fading**	0.00	1.61	0.00	0.22	0.00	0.00	**0.88**
54	acacia	random	federal	0.00	0.00	0.00	0.07	0.11	0.69	0.16
55	acacia	random	greeting	0.00	0.16	0.00	0.15	0.05	0.00	0.19
56	acacia	random	hope	0.00	0.21	0.00	0.29	0.00	0.69	0.17
57	acacia	random	impact	0.00	0.00	0.01	0.10	0.00	0.23	0.09
58	acacia	random	important	0.00	0.57	0.01	0.38	0.00	0.00	0.61
59	acacia	random	instrument	0.00	0.00	0.01	0.09	0.00	0.69	0.10
60	acacia	random	knock-on	0.00	0.00	0.00	0.01	0.00	0.00	0.11
61	acacia	random	learn	0.00	0.00	0.00	0.23	0.00	0.00	0.34
62	acacia	random	manuscript	0.00	0.00	0.01	0.40	0.00	0.00	0.24
63	acacia	random	mezzo-soprano	0.00	0.00	0.00	0.08	0.00	0.69	0.07
64	acacia	random	moor	0.00	0.73	0.04	0.45	0.01	0.69	0.67
65	acacia	random	mutual	0.00	0.18	0.00	0.05	0.00	0.00	0.10
66	acacia	random	nappy	0.00	0.00	0.00	0.02	0.01	0.52	0.09
67	acacia	random	natural	0.00	0.00	0.15	0.41	0.05	0.69	0.54
68	**acacia**	**random**	**open**	0.00	0.68	0.07	1.37	0.07	0.44	**1.88**
69	acacia	random	re	0.00	0.28	0.00	0.05	0.07	0.23	0.14
70	acacia	random	responsible	0.00	0.12	0.00	0.17	0.00	0.00	0.36
71	acacia	random	show	0.00	0.00	0.01	0.31	0.00	0.00	0.18
72	acacia	random	stumble	0.00	0.00	0.05	0.28	0.00	0.00	0.21
73	acacia	random	sycophantic	0.00	0.00	0.00	0.00	0.00	0.00	0.02
74	acacia	random	tour	0.00	0.00	0.03	0.22	0.00	0.00	0.21

Table 4.5: Similarity scores between terms related to the word "acacia" in the BLESS dataset calculated by PatternSim-Efreq, LSA-Tasa, BDA-3-Cos, SDA-21-Cos, DefVectors-WktWiki, Resnik and Logit-E15 measures. The original scores were transformed into z-scores. The most distinct false positives and false negatives of Logit-E15 are highlighted in bold.

Chapter 5

Applications of Semantic Similarity Measures

"Essentially, all models are wrong, but some are useful."

– George E. P. Box

This chapter presents two applications of semantic similarity measures to natural language processing. First, Section 5.1 presents *Serelex*, a system based on a semantic similarity measure. Given a query, this system provides a list of related terms and displays them as a list, as an interactive graph or as a set of images. Second, in Section 5.2 we present a system processing filenames on P2P networks. We show that the relations automatically extracted with a similarity measure improve classification accuracy with help of the *vocabulary projection* technique. Finally, in Section 5.3 we provide a list of text processing applications, where semantic similarity measures may be potentially useful.

We conclude that the presented semantic similarity measures indeed can be practical in the real language processing systems. Both systems described in this chapter rely on the *PatternSim* measure (see Section 2.4). In future work, it would be useful to integrate into these applications more advanced measures, such as *C-SVM-radial-E15* (see Chapter 4).

5.1 Serelex: Search and Visualization of Semantically Similar Words[1]

We present *Serelex*, a system that, given a query in English, returns a list of related terms ranked according to a *semantic similarity measure*. The system helps to learn the meaning of

[1]The research presented in this section was published as Panchenko et al. [2] and Panchenko et al. [3].

a query term and to discover semantically similar words in an interactive way. Some systems such as Visual Thesaurus [2], VisuWords [3], VisGloss [4], Visual Synonyms[5] or VisualWorld [6] show a list of related terms in a form of a graph (see Figure 5.1). Our system also implements such a visualization.

Unlike thesauri-based systems (e. g. `Thesaurus.com`, WordNet or Visual Synonyms), *Serelex* relies on information extracted from text corpora. In contrast to systems based on associative tests (e. g., JeuxDeMots [7], Lexfn [8], Edinburgh Associative Thesaurus [9] or Russian Associative Thesaurus [10]), our system do not need any human judgment. In comparison to other similar systems (e. g., BabelNet [11], ConceptNet [12], UBY [13]), *Serelex* does not depend on a semantic resource like WordNet. Instead, we build upon an original pattern-based similarity measure described in Section 2.4, which extracts semantic relations from texts. The proposed system has a precision rate comparable to those of the baselines. Furthermore, it has a larger lexical coverage than the dictionary-based systems, provides list-, graph-, and image-based GUIs, and is open source. Last but not the least, it can be automatically updated with relations extracted from new documents.

5.1.1 The System

Serelex is freely available online [14]. Figure 5.2 presents its structure, which consists of an extractor, a server and a user interface. The extractor gathers semantic relations between words from a raw text corpus. The extraction process occurs offline. The extracted relations are stored in the database. The server provides fast access to the extracted relations over HTTP. A user interacts with the system through a web interface or an API. The system as well as the data and evaluation scripts are open source [15].

[2] `http://www.visualthesaurus.com/`
[3] `http://www.visuwords.com/`
[4] `http://visgloss.com/`
[5] `http://www.visualsynonyms.com/`
[6] `http://visualworld.ru/`
[7] `http://www.jeuxdemots.org/`
[8] `http://www.lexfn.com/`
[9] `http://www.eat.rl.ac.uk/`
[10] `http://tesaurus.ru/dict/dict.php`
[11] `http://lcl.uniroma1.it/bnxplorer/`
[12] `http://conceptnet5.media.mit.edu/`
[13] `https://uby.ukp.informatik.tu-darmstadt.de/webui/tryuby/`
[14] `http://serelex.cental.be`
[15] `http://serelex.cental.be/page/about`, available under conditions of LGPLv3 license.

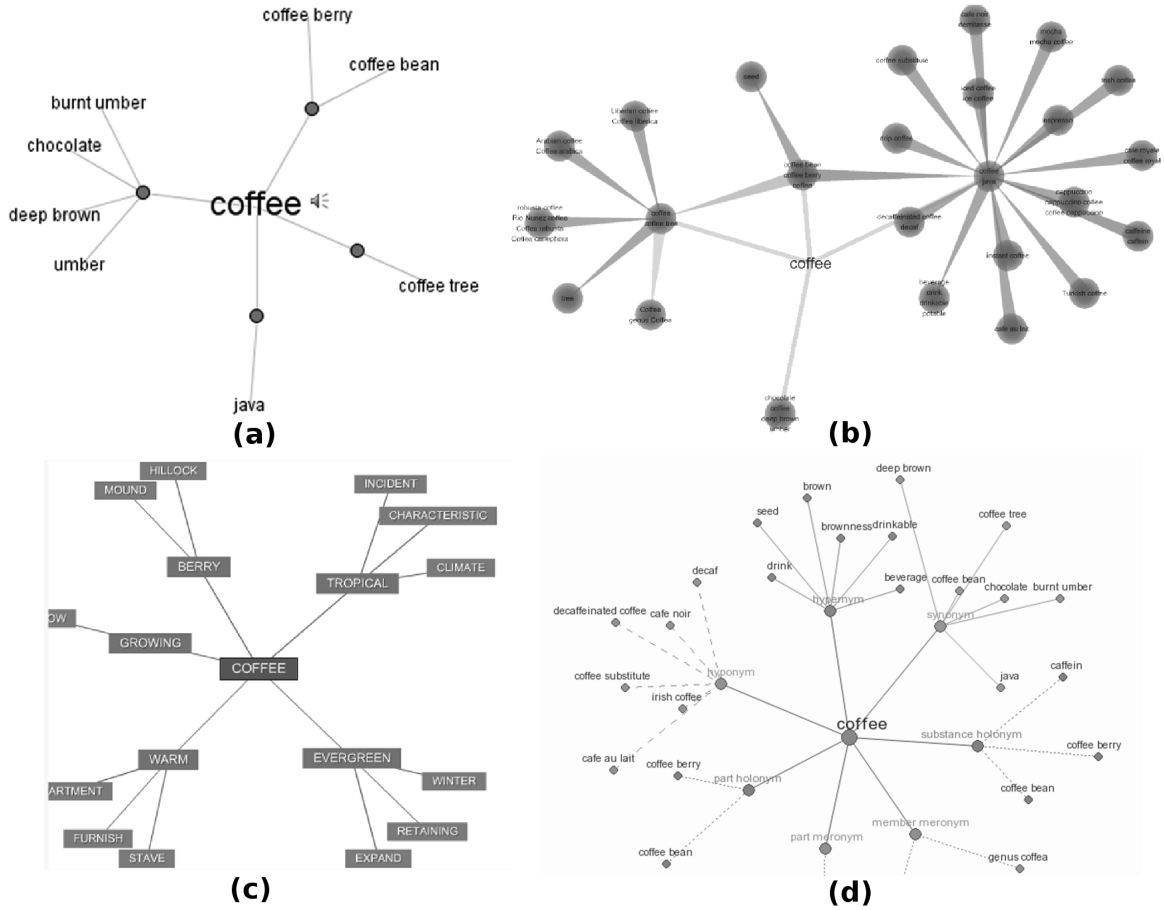


Figure 5.1: Semantic relations of the term "coffee" visualized by (a) Visual Thesaurus, (b) VisuWords, (c) VisGloss and (d) Visual Synonyms.

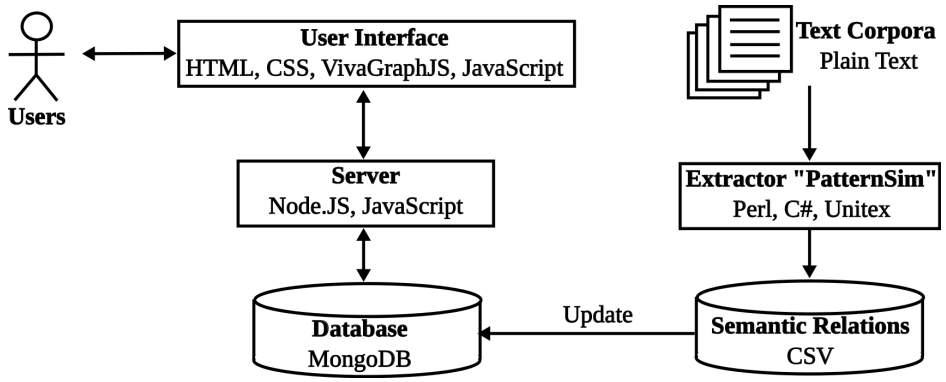


Figure 5.2: Structure of the "Serelex" system.

Extractor

The extractor is based on the semantic similarity measure *PatternSim* and *Efreq-Rnum-Cfreq-Pnum* re-ranking formula (see Section 2.4). We used as a corpus a combination of Wikipedia abstracts and ukWaC (Baroni et al., 2009) (5,387,431 documents, $2.915 \cdot 10^9$ tokens, 7,585,989 lemmas, 17.64 Gb). Processing of the corpus took around 72 hours on a standard machine (Intel i5, 4Gb RAM, HDD 5400rpm). The result of the extraction is

11,251,240 untyped semantic relations, such as $\langle Canon, Nikon, 0.62 \rangle$, between 419,751 terms.

Server

The server returns a list of related words for each query, ranked according to their semantic similarity score stored in the database. The queries are lemmatized with the DELA dictionary [16]. An approximate search is performed for queries with no results. The system can import networks in CSV format created by other similarity metrics and extractors.

User Interface

One can access the system via a graphical user interface or a RESTful API. The GUI consists of three key elements: a search field, a list of the results and a graph of the results (see Figures 5.3 and 5.4). A user interacts with the system by issuing a query – a single word such as "mathematics" or a multiword expression such as "computational linguistics". Query suggestions are sorted at the same time by term frequency in the corpus, by query frequency and alphabetically. A list of results contains 20 terms which are the most semantically related to the query.

The graph of results provides an alternative representation of the toplist. It enables visualization of semantic relations with a force-directed graph layout algorithm based on the Barnes-Hut simulation (Barnes and Hut, 1986). The layout incorporates the secondary relations: words related to the words linked to the query. This lets the layout algorithm cluster the results. For instance, Figure 5.3 clearly demonstrates that the term "jaguar" may be related either to cars or to animals. Similarly, the layout of term "python" lets a user seamlessly identify the two meaning of this word. Therefore, the graph provides a natural way to plot dense communities of related terms. Furthermore, the graph layout structures the results and thus lets to a user perceive more than 20 results at once.

The system can distinguish even between somewhat more fine-grained word senses. For example, Figure 5.4 groups Belgian cities in one cluster (Brussels, Liege, Ghent, Bruges, Antwerp and Charleroi) and the World capitals in another one (Brussels, Amsterdam, Berlin, Madrid, Copenhagen, Beijing, Buenos Aires, etc.). It is possible to detect dense communities in the graph explicitly with various graph clustering algorithms (Dhillon et al., 2004; Yen et al., 2007; Schaeffer, 2007; Blondel et al., 2008; Fortunato, 2010).

The graph of results is interactive. A user can issue additional queries by clicking on the

[16] `http://infolingu.univ-mlv.fr/`, available under conditions of LGPLLR.

nodes. In this case, a new search query is generated and its results are added to the initial graph (see Figure 5.5). In this way, a user may use the GUI to identify and develop clusters of strongly related words. For instance, the graph represented in Figure 5.5 contains clusters of Indonesian islands (activated by the queries “malaya”, “borneo” and “indonesia”), programming languages (activated by the queries “java”, “php”, “perl”, “ruby” and “python”), reptiles (activated by the queries “alligator”, “snake” and “crocodile”) and precious stones (activated by the queries “ruby”, “topaz”, “garmet” and “tourmaline”). Note that the ambiguous words (“java”, “ruby” and “python”) are the articulation points of this graph.

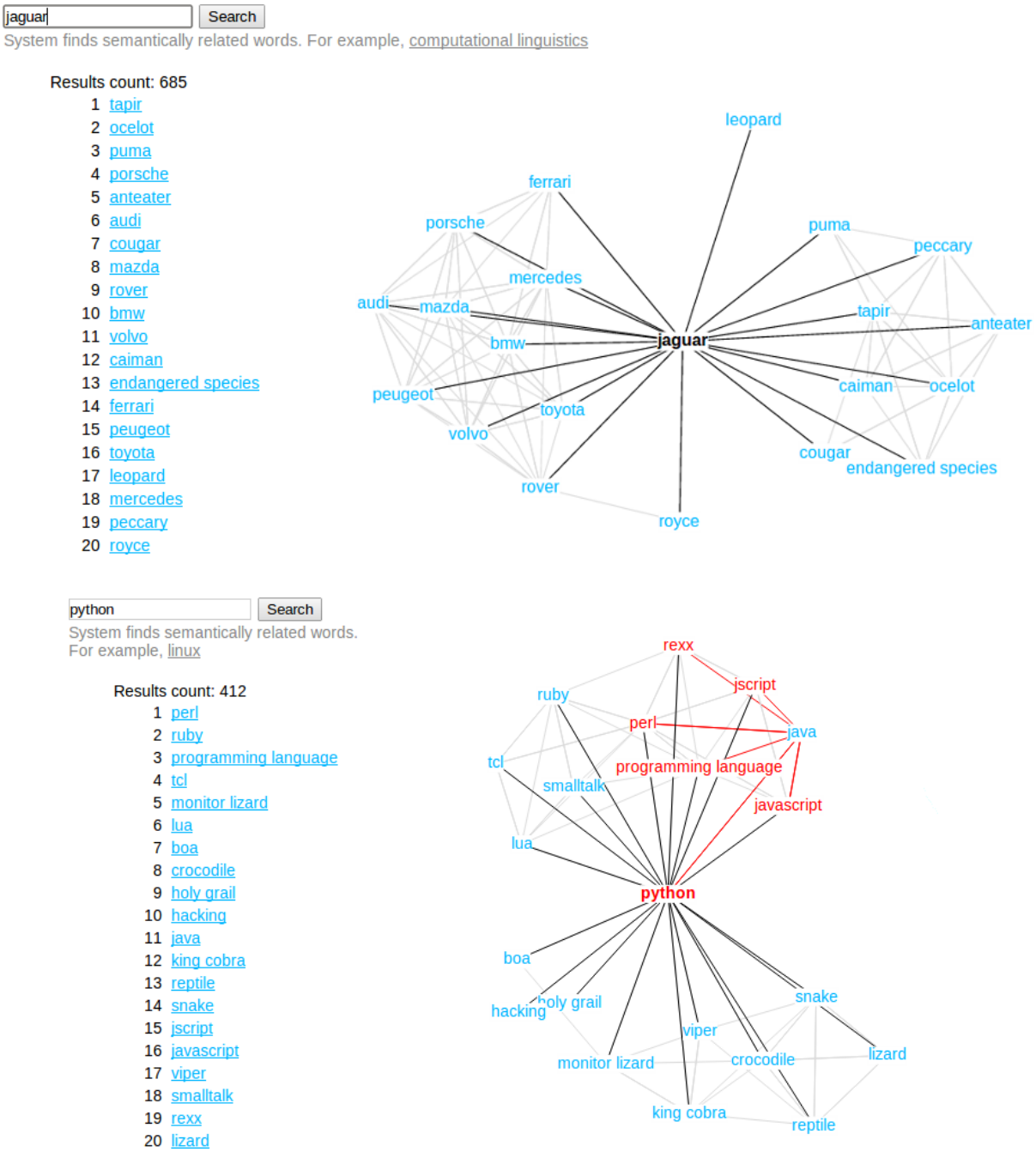


Figure 5.3: Graphical user interface of the lexico-semantic search engine “Serelex”.

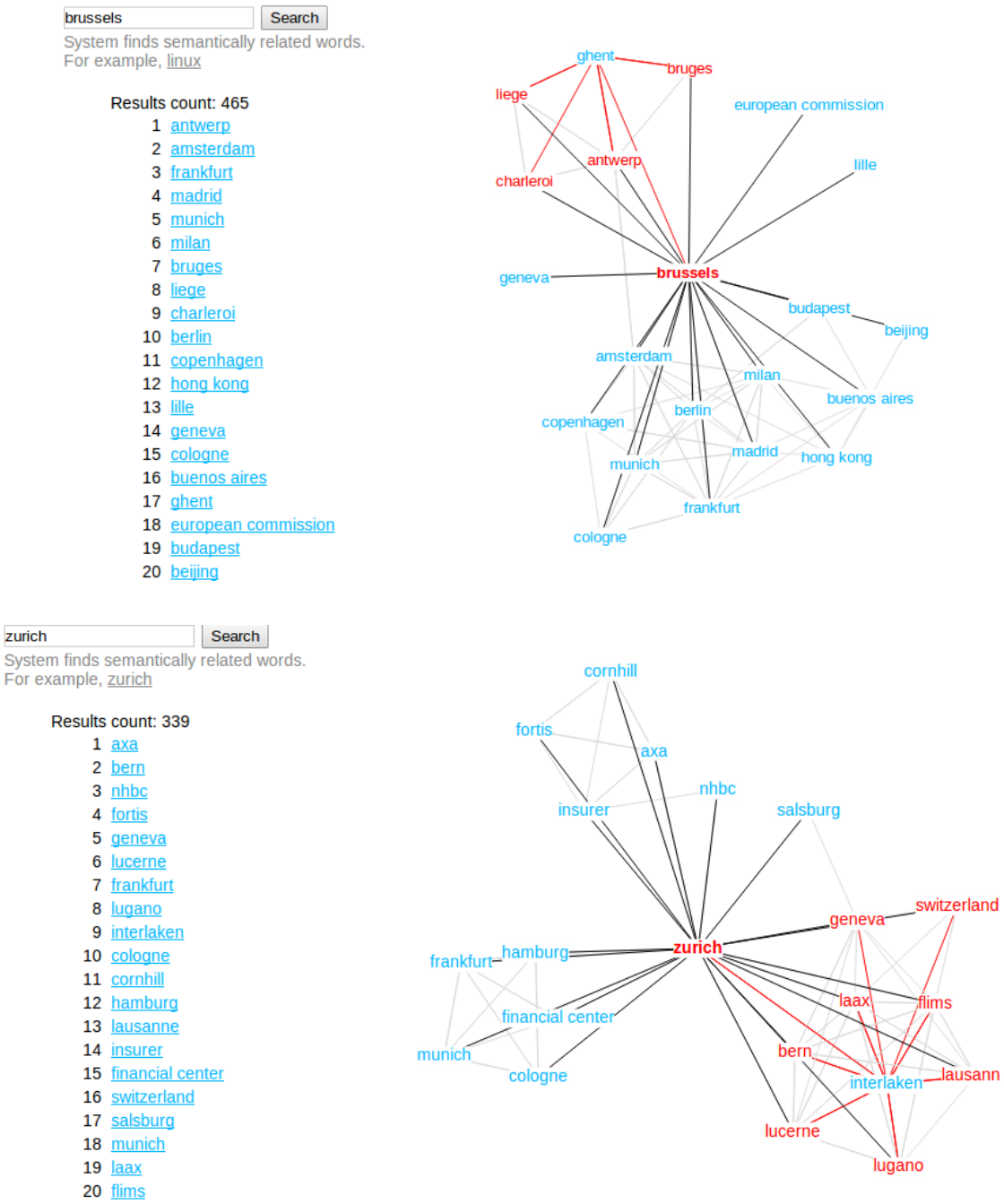


Figure 5.4: Graphical user interface of the lexico-semantic search engine "Serelex".

The system can also visualize the results as a set of images (see Figure 5.6). In this case, the search results are represented with images from the Google Image Search [17]. This visualization allows a user to perceive related words more quickly and intuitively. Additional examples of this visualization mode are provided in Appendix A. The appendix also describes the Serelex applications for the Microsoft Windows platform.

[17] In this prototype, we used a wrapper around the Google Images: `http://jpg.to/`.

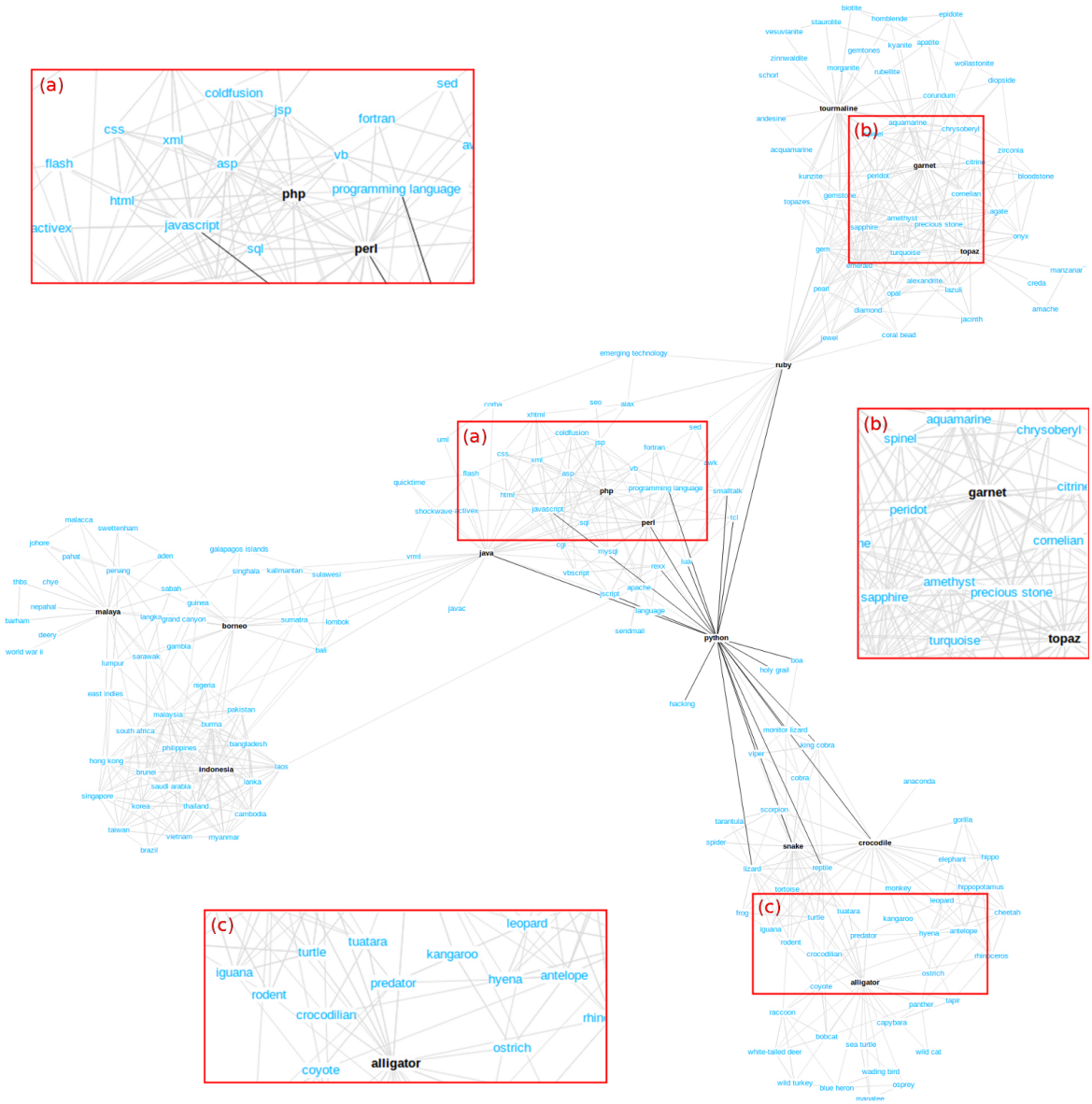


Figure 5.5: Interaction with the graph of results let user identify clusters of related words. Here "python" is the initial query and the other black nodes are the secondary queries.

5.1.2 Evaluation and Results

We evaluated the system against four tasks: correlations with human judgments, semantic relation ranking, extraction of semantic relations and user satisfaction (see Section 1.2.3). The first three tasks are extrinsic evaluations of the similarity measure *PatternSim*. Results of these benchmarks were presented in Section 2.4. Here we only recall the main points relevant to the *Serelex* system.

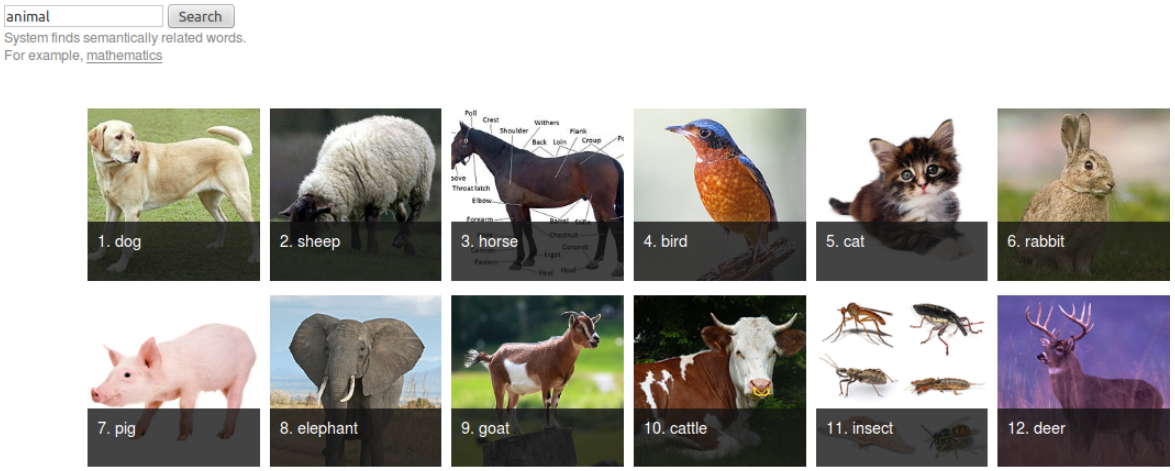


Figure 5.6: Serelex: search results for the query "animal" visualized with images from Google (see Appendix A for additional examples).

Correlation with Human Judgements and Semantic Relation Ranking

According to the correlations with human judgements (see Section 1.2.3), the system performs comparably to the baseline measures based on WordNet (*WuPalmer, LecockChodorow*, *Resnik*), corpora (*BDA, SDA, LSA*) and definitions (*DefVectors-WktWiki, GlossVectors, ExtendedLesk*). Similar results were obtained on the semantic relation ranking task. In terms of precision, it outperforms 9 mentioned above baselines, but its recall is seriously lower than those of baselines. Thus, output of the system is consistent with the common notions of semantic similarity.

Semantic Relation Extraction

We estimated the average precision of the extracted relations for 49 queries. It varies between 0.736 for the top relation and 0.599 for the top 50 relations (see Figure 5.7 (a)). This information may be useful if one would like to use output of *Serelex* in some NLP pipeline.

User Satisfaction

We also measured user satisfaction with our results. 23 assessors were asked to issue 20 queries of their choice and, for each of them, to rank the top 20 results as relevant, irrelevant or a mix of both. We collected 460 judgements from the 23 assessors and 233 judgements from 109 anonymous users (see Figure 5.7 (b)). Users and assessors (users asked to assess the system) issued together 594 distinct queries. According to this experiment, the results are relevant in 70% of the cases and irrelevant in 10% of the cases. Finally, 20% of queries recall both relevant and irrelevant results.

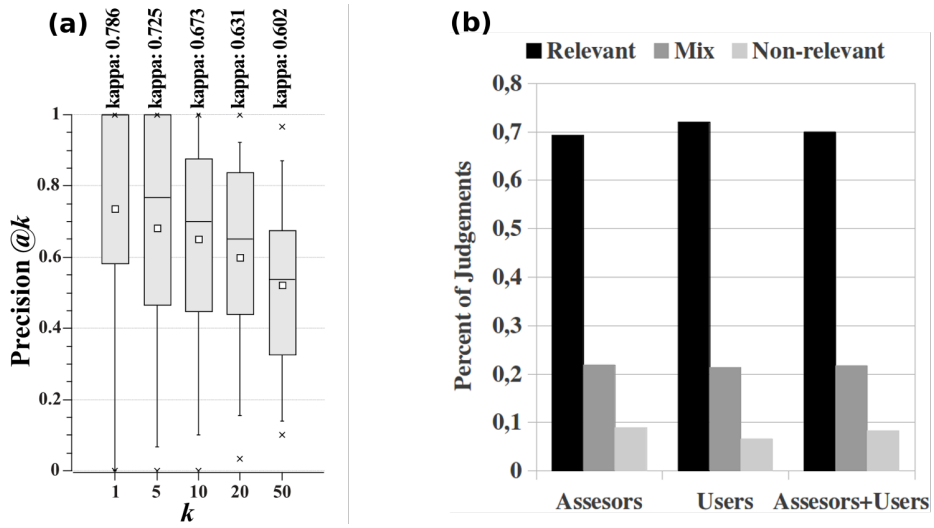


Figure 5.7: Evaluation: (a) semantic relation extraction task; (b) users' satisfaction of top 20 results.

5.1.3 Summary

We presented a system which finds semantically related words. Our results have shown that its precision is comparable to the dictionary-based baselines and a better coverage as it extracts relations directly from texts. The system achieves a Precision@1 of around 74% and users are satisfied with 70% of the query results. Most importantly, the system does not need any manually-crafted dictionary to achieve these results.

5.2 Short Text Categorization[18]

This section presents a system for filename categorization, which was designed to identify pedophile media files on the P2P networks by their textual descriptions. In our initial experiments, we used regular pornography data as a substitution of child pornography.

The goal of the iCOP project[19] is to develop a toolkit which helps law enforcement agencies across the EU identify child sexual abuse (CSA) media and its originators on P2P networks. Until now, the usual way to identify such media was through manual analysis. Such an approach is impractical as thousands of new files appear every day. We describe a text processing module of iCOP, designed to recognize the CSA media by their filenames. These media are further prioritized with a content-based media analysis (Ulges and Stahl, 2011) and a user behaviour analysis.

Contributions of this section are the following. First, we present two datasets which can

[18]The research presented in this section was published as Panchenko et al. [1] and Panchenko et al. [4].

[19]The project is funded by EU Safer Internet Programme "Empowering and Protecting Children Online" under contract SI-2601002: `http://ec.europa.eu/information_society/activities/sip`, `http://scc-sentinel.lancs.ac.uk/icop/`.

be used to train/test filename classifiers. Second, we perform a computational study of various approaches to filename classification. Finally, we present an open source system for short text categorization, which recognizes pornographic filenames with an accuracy rate up of 91–97%. It implements an original *vocabulary projection* technique, which helps to overcome vocabulary mismatch.

5.2.1 Related Work

Text categorization is a task that received much attention in the literature and robust methods has been developed (Sebastiani, 2002; Joachims, 1998). There exist two main approaches to text categorization: symbolic and statistical (Yang and Liu, 1999; Sebastiani, 2002; Ageev et al., 2008). But filename categorization is a special case of *short* text categorization (Sriram et al., 2010). This task is challenging as filenames may be very short and/or meaningless especially those of CSA media.

Recent research on cyberpedophilia has been focused on chat analysis. Pendar (2007) have built a system that separates a predator and a victim based on chats from the Perverted Justice site [20]. McGhee et al. (2011) have used the same dataset to classify chat lines. Bogdanova et al. (2012) have built a system which identifies pedophiles among chat users based on emotion-based features from the same chat data. Peersman et al. (2011) have built a system for age detection in chats based on the Netlog corpus (Kestemont et al., 2012). A shared task "Sexual Predator Identification" has been introduced at PAN'12 (Kontostathis et al., 2012; Peersman et al., 2012). However, to the best of our knowledge, identification of CSA media based on its text description was not yet investigated.

Our system relies on a statistical text classifier refined with semantic relations extracted from a text corpus. Some researchers already tried to incorporate semantic resources in text categorization systems. Dobrov, Loukachevich and Ageev (Dobrov, 2002; Loukachevitch et al., 2002; Ageev et al., 2008) developed a knowledge-based text categorization method, which relies on a thesaurus. In their approach, a category is defined as a conjunction of disjunctions of several terms. For instance, the category *home repair* is defined as follows: $(\textit{repair} \vee \textit{complete overhaul} \vee \textit{light repair} \vee \textit{repair-and-renewal operations}) \wedge (\textit{residential building} \vee \textit{living space} \vee \textit{flat})$. This definition is automatically expanded with help of the semantic relations of the thesaurus. The method outperforms the state-of-the-art statistical classifiers when little training data available or when training data are not consistent. To support their claim, the authors classified documents against complex hierarchical

[20] `http://www.perverted-justice.com/`

schemes including Legislative Indexing Vocabulary [21], DMOZ [22] and Legislative Classifier of Russian Federation. Kevers (Kevers, 2009; Kevers et al., 2011) proposed a method which builds a text classifier from a thesaurus. The method builds a cascade of finite-state transducers for each category from a set of terms describing this category. The transducer recognizes descriptors in a document, taking into account several types of linguistic variations. The method performs comparably with the statistical state-of-the-art techniques and outperforms them when a little number of training documents is available. Tikk et al. (2003) proposed to incorporate into a relations of a thesaurus into a k-NN classifier. The method improves the precision of the standard k-NN approach up to 13% without compromising its recall.

5.2.2 Filename Classification

The file classification module [23] is designed to recognize pedophile media based on textual descriptions of the associated files. The module consists of a feature extractor and a classifier trained on a specific dataset.

Feature Extraction

First, the text associated to a file (title, tags and description) is cleaned up from special symbols and tokenized. Next, the filename is lemmatized with *TreeTagger* (Schmid, 1994). The standard stopwords are removed with the exception of the "sex-related" ones such as "him", "her", "woman", etc. Finally, a file is represented as a unit length bag-of-words vector of lemmas.

Filenames usually contain only a few meaningful words. If none of them matches the vocabulary of a pre-trained classifier, then classification is not possible. We address this issue with the *vocabulary projection*. This technique projects an out-of-vocabulary lemma into the vocabulary of a classifier with the help of 11,251,240 semantic relations over 419,751 terms learned from a text corpus with the *PatternSim* semantic similarity measure (Panchenko et al., 2012). This procedure, for each out-of-vocabulary lemma w looks up $n \in [10; 20]$ most semantically similar words. Related words which are present in the vocabulary of the classifier are added to the bag-of-words vector instead of w. Normally, an out-of-vocabulary word is replaced with its in-vocabulary synonym, hypernym or co-hyponym. However, erroneous expansions of very short texts may lead to a wrong prediction.

[21] `http://thomas.loc.gov/liv/livtoc.html`
[22] Open Directory Project: `http://www.dmoz.org/`.
[23] `http://github.com/cental/stc`, available under conditions of LGPLv3.

Figure 5.8 presents some examples of the vocabulary projection [24]. In this example, a classifier is trained on the *Gallery* dataset and tested on the *PirateBay* dataset (see below). Here each example depicts a tokenized filename and features extracted from it, e.g.:

```
Angy with Katrine 21 03 2012 wmv
>>angy (@)
>>with (-)
>>katrine (17) = velda, trossachs, ursula, watercourse, katherine, nan,↩
   ↩ tara, jasmine, helena, outing, loch, douglas.
>>21 (-)
>>03 (-)
>>2012 (-)
>>wmv (3) = dat; rm; standalone;
```

Features marked with the `(@)` symbol are in-vocabulary words. They are present in the vocabulary of a classifier. Features marked with the `(-)` symbol are stop words. Features marked with the (x) symbol, where $x \geq 0$ are the out-of-vocabulary terms. Here x is the number of successfully projected terms from the out-of-vocabulary term to the vocabulary of the classifier. In the example presented below, the out-of-vocabulary term `katrine` is represented with several semantically similar in-vocabulary terms: `velda, ursula, katherine`, and so on. The projected terms are often not precise synonyms of the out-of-vocabulary word. Furthermore, the projection may be wrong. However, this technique lets the classifier receive at least some information about an out-of-vocabulary term.

Classification

Binary linear statistical classifiers are used to separate regular files from those containing pornographic content. We experimented with Support Vector Machines (Section 4.2), Regularized Logistic Regression (Section 4.2), Perceptron (Theodoridis and Koutroumbas, 2009, p.93) and Least Mean Squares algorithm (Theodoridis and Koutroumbas, 2009, p.105) [25]. To avoid overfitting, we used cross-validation which extracts vocabulary of the classifier only from the training fold.

Training Data

In our system, positive training examples are texts associated with pornographic resources, while negative training examples are texts associated with non-pornographic content. In the experiments described here, "regular" pornographic filenames are used as a substitute

[24] All examples from this experiment are available from: `http://cental.fltr.ucl.ac.be/team/panchenko/thesis/voc-projection.log`.

[25] We used *LibSVM* (Chang and Lin, 2011), *LibLINEAR* (Fan et al., 2008) and MATLAB.

```
mokey30 IRIX
>>mokey30 (-)
>>irix (5) = solaris, freebsd, nix, bsd, tru.

Plan9 unix
>>plan9 (-)
>>unix (9) = linux macintosh, solaris, freebsd, bsd, window, platform, ↩
   ↩novell, sco.

Darkman 1990 BRRip
>>darkman (8) = zombie, iii, indian, die, soldier, character, film.
>>1990 (-)
>>brrip (0)

Tomtom map Europe 865
>>tomtom (12) = mio, amro, nobel, cobra, kane, continental, horizon, ↩
   ↩gps, delta, amsterdam, mapping, sony.
>>map (@)
>>europe (@)
>>865 (-)

OneManBanned 07 07 08 Chelsea Sinclaire Hardcore XXX
>>onemanbanned (0)
>>07 (-)
>>07 (-)
>>08 (-)
>>chelsea (@)
>>sinclaire (13) = kener, raylene, dasha, jameson, dayton, hayes, ↩
   ↩richardson, turner, performer, silver, cast, bank, star.
>>hardcore (@)
>>xxx (@)

Violet and Labrn furry Pleasure Bonbon
>>violet (@)
>>and (-)
>>labrn (0)
>>furry (@)
>>pleasure (@)
>>bonbon (7) = geoffrey, candy, cake, sweet, bottle, dish, suite.
```

Figure 5.8: Examples of the vocabulary projection (the maximum number of expansions $n = 30$). The classifier was trained on the Gallery dataset and tested on the PirateBay Titles+Tags dataset.

for child pornography filenames. First, such data share important characteristics with CSA material like sex-related vocabulary, file extensions, etc. Indeed, CSA is a special case of pornographic data. Second, CSA data were not yet provided by our law enforcement collaborators. Thus, we constructed ourselves two datasets from openly available data: *Gallery* and *PirateBay*.

The *Gallery* dataset contains 106,350 texts. Positive examples of this dataset were gathered

from four sites: PicHunter, PornoHub, RedTube and Xvideos [26]. Each of 51,350 positive training examples is composed of a title and tags of a porn video or a porn gallery, e. g.:

- `Beautiful girl in hardcore action`,
- `Slim Can Bearly Take The Dick`.

Negative training examples in this dataset are 55,000 randomly selected titles from the English Wikipedia, each composed of at least 4 words e. g.:

- `Contractors and General Workers Trade Union`,
- `1957-58 American Hockey League season`.

The *PirateBay* dataset consists of data available from ThePirateBay torrent tracker [27]. The files of this tracker are organized in six main categories such as "applications" or "porn" and 39 subcategories such as "applications-android" or "porn-movies". We crawled titles, tags and descriptions of 100,064 files from all categories. So each sub-category is represented with around 3,000 files. From this material, we constructed a dataset of 16,863 positive training examples (the porn category) and 83,201 negative training examples (the five other categories). We constructed two versions of this dataset. The *PirateBay-TT* includes *T*exts and *T*ags associated with the files, while the *PirateBay-TDT* consists of *T*exts, *D*escriptions and *T*ags.

5.2.3 Evaluation and Results

Our results presented in the first part of the Table 5.1 suggest that a Support Vector Machine or a Logistic Regression clearly distinguishes a pornographic text from a non-pornographic one. In particular, the best results on the *Gallery* dataset (96.52%) are obtained by a C-SVM with the linear kernel. Figure 5.9 (b) depicts results of the metaparameter optimization of this model with the grid search. As one can see, this procedure improves the accuracy only by 0.4%. Such a small variance of the model is useful for our application as the system should be automatically retrained by the Police.

The second part of the Table 5.1 reports on performance of the best model trained and applied to different datasets. It appears that, the classifier is able to correctly model both *Gallery* and *PirateBay* datasets. Furthermore, the model does not seem to be particularly overfitted. Accuracy of the classifier trained on the *Gallery* dataset and applied on the *PirateBay* dataset and vise-versa achieves up to 91%. Figure 5.9 presents further information about the classifier trained on the *Gallery* dataset and tested on the *PirateBay* dataset. The two constantly misclassified sub-categories are "other-other" and "porn-games". Filenames

[26] `www.pichunter.com`, `www.porno-hub.com`, `www.redtube.com`, `www.xvideos.com`
[27] `http://thepiratebay.org/`

Model	Training Dataset	Test Dataset	Accuracy	Accuracy (voc.proj.)
C-SVM, linear kernel	Gallery	Gallery	**96.52**	–
Logistic Regression (L2-reg.)	Gallery	Gallery	96.27	–
Perceptron ($\epsilon \leq 1\%$, 570 iter.)	Gallery	Gallery	94.03	–
Logistic Regression (L1-reg.)	Gallery	Gallery	93.95	–
Least Mean Squares ($\rho = 10$)	Gallery	Gallery	91.85	–
ν-SVM, radial kernel	Gallery	Gallery	88.35	–
ν-SVM, linear kernel	Gallery	Gallery	88.20	–
ν-SVM, sigmoid kernel	Gallery	Gallery	87.45	–
ν-SVM, polynomial kernel	Gallery	Gallery	79.77	–
C-SVM, polynomial kernel	Gallery	Gallery	51.71	–
C-SVM, radial kernel	Gallery	Gallery	51.71	–
C-SVM, sigmoid kernel	Gallery	Gallery	51.71	–
C-SVM, linear kernel	Gallery	Gallery	96.52	96.83 (+0.42)
C-SVM, linear kernel	Gallery	PirateBay-TDT	90.57	91.48 (+0.91)
C-SVM, linear kernel	Gallery	PirateBay-TT	84.23	88.89 **(+4.66)**
C-SVM, linear kernel	PirateBay-TT	Gallery	91.16	91.30 (+0.14)
C-SVM, linear kernel	PirateBay-TT	PirateBay-TT	97.73	97.63 (-0.10)
C-SVM, linear kernel	PirateBay-TDT	Gallery	88.83	89.04 (+0.21)

Table 5.1: Performance of different binary filename classifiers (10-fold cross-validation).

of the latter are indeed difficult to classify as they are similar to those of video games (e.g. "3D SexVilla Crack").

According our experiments summarized in Figure 5.9 and Table 5.1, training a model on the noisy descriptions of the *PirateBay-TDT* hampers accuracy of the classifier by around 3%. On the other hand, using those descriptions at the classification time provides an improvement up to 6%. Finally, the *vocabulary projection* indeed helps to deal with the vocabulary mismatch issue. It improves accuracy of a classifier trained on *Gallery* and tested on *PirateBay-TT* by 4.66%.

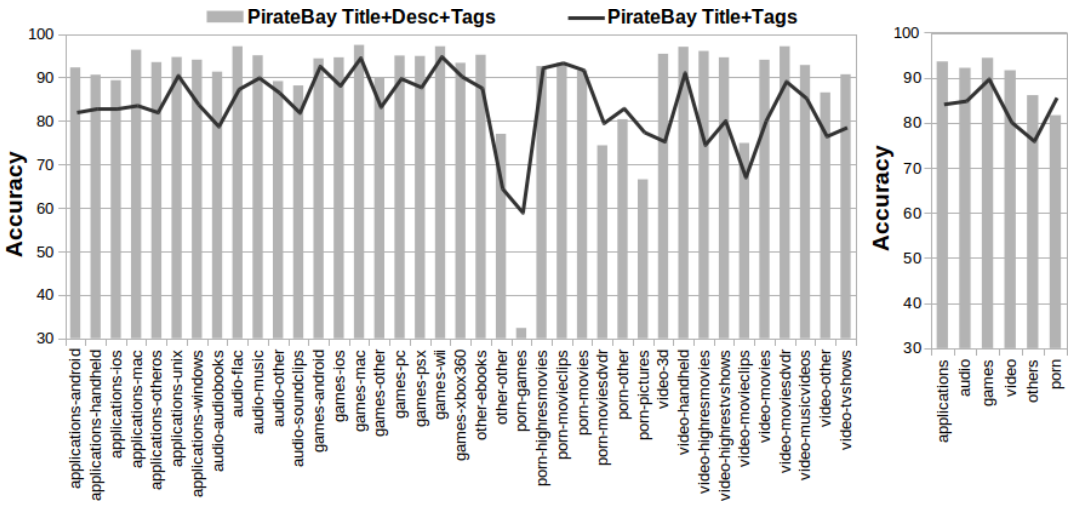


Figure 5.9: C-SVM-linear trained on the Gallery dataset and tested on the PirateBay dataset.

5.2.4 Examples of the Vocabulary Projection

The vocabulary projection technique appeared to be most useful for the classifier trained on the *Gallery* dataset and applied to the *PirateBay Titles+Tags* dataset (see Table 5.1). In this case, the technique improved accuracy by 4.66% or by 4,026 texts. We analysed two cases: the one without vocabulary projection (accuracy of 84,23%) and the one with it (accuracy of 88,89%). In this comparison, the number of expansions n was set to 30 words [28]. For the first case, the number of the true classifications T is equal to

$$T = \#(TP \cup TN) = 75,857, \tag{5.1}$$

where TP is a set of true positives and TN is a set of true negatives. For the second case, the number of true classifications T' is equal to

$$T' = \#(TP' \cup TN') = 79,883. \tag{5.2}$$

Note that in general $TP' \not\supseteq TP$ and $TN' \not\supseteq TN$, i. e. the projection may flip both correct and incorrect classifications. Indeed, we have observed all the four cases:

1. $x \to y, x \in FN, y \in TP'$ – a corrected false negative, where FN is a set of false negatives, e. g.:

```
18XGirls Yulia
>>18xgirls (-)
>>yulia (3) = ekaterina, sonya, daughter.

Sexart 12 04 05 Nedda A Presenting Nedda SexArt
>>sexart (0)
>>12 (-)
>>04 (-)
>>05 (-)
>>nedda (9) = adina, gilda, mimi, juliette, marguerite, heroine, ↩
   ↩lucia, liu, role.
>>a (-)
>>present (@)
>>nedda (9) = adina, gilda, mimi, juliette, marguerite, heroine, ↩
   ↩lucia, liu, role.
>>sexart (0)
```

2. $x \to y, x \in TP, y \in FN'$ – a miscorrected true positive, e.g.:

```
Violated Heroin Violated Heroin
```

[28] All texts classified in this experiment are available from: `http://cental.fltr.ucl.ac.be/team/panchenko/thesis/voc-proj.tgz`

```
>>violate (@)
>>heroin (8) = cocaine, ecstasy, lsd, cannabis, marijuana, opium, ↩
   ↩narcotic, crack.
>>violate (@)
>>heroin (8) = cocaine, ecstasy, lsd, cannabis, marijuana, opium, ↩
   ↩narcotic, crack.

Chanel Preston My Naughty Massage Copy
>>chanel (@)
>>preston (@)
>>my (-)
>>naughty (@)
>>massage (@)
>>copy (18) = paste, artwork, headline, document, documentation, ↩
   ↩cut, material, delete, photograph, text, print, postage, ↩
   ↩paper, command, braille, file, correspondence, glossy.
```

3. $x \rightarrow y, x \in FP, y \in TN'$ – a corrected false positive, where FP is a set of false positives, e. g.:

```
HD Widgets Android
>>hd (@)
>>widget (3) = gadget, menu, button.
>>android (@)

iMovie for iPhone 3GS (IOS4) IOS4 iPHone iMovie
>>imovie (13) = itunes, nero, pinnacle, premiere, maker, footage, ↩
   ↩explorer, express, apple, application, studio, software.
>>for (-)
>>iphone (@)
>>3gs (-)
>>( (-)
>>ios4 (-)
>>) (-)
>>ios4 (-)
>>iphone (@)
>>imovie (13) = itunes, nero, pinnacle, premiere, maker, footage, ↩
   ↩explorer, express, apple, application, studio, software.
```

4. $x \rightarrow y, x \in TN, y \in FP'$ – a miscorrected true negative, e. g.:

```
xbmc android app xbmc
>>xbmc (2) = dashboard, boot.
>>android (@)
>>app (2) = tron, gimp.
>>xbmc (2) = dashboard, boot.
```

```
gparted 0 3 4 2
>>gparted (1) = tool.
>>0 (-)
>>3 (-)
>>4 (-)
>>2 (-)
```

No features can be extracted from some filenames without the vocabulary projection, e. g.:

```
bw065 thr8 avi
>>bw065 (-)
>>thr8 (-)
>>avi (3) = png, tiff, psd.

18XGirls Yulia
>>18xgirls (-)
>>yulia (3) = ekaterina, sonya, daughter.

Bunheads S01E03 HDTV XviD AFG
>>bunheads (0)
>>s01e03 (-)
>>hdtv (5) = vod, hbo, dab, playback, ipod.
>>xvid (12) = rm, format, digital, mp, disc, ac, movie, file, video, ↩
   ↩library, type, product.
>>afg (0)
```

Most of the filenames contain only a couple of keywords, such as "Adobe Photoshop", "DosBox" or "TomTom". The other terms refer usually to the data format (e. g. "XviD", "avi" or "HD 720p"), the file date/version (e. g. "2012 06 21"), or to the user uploaded the file (e. g. "ezir", "NLTorrents" or "JohnPc666"). The projection is especially useful if it recovers a missing key term, as in the examples below:

```
TomTom Eastern Europa v1 10 Iphone Ipad
>>tomtom (12) = mio, amro, nobel, cobra, kane, continental, horizon, ↩
   ↩gps, delta, amsterdam, mapping, sony.
>>eastern (@)
>>europa (@)
>>v1 (-)
>>10 (-)
>>iphone (@)
>>ipad (0)

Woody Woodpecker ipod ipad iphone
>>woody (@)
>>woodpecker (8) = tit, owl, buzzard, warbler, oriole, squirrel, jay, ↩
   ↩wren.
```

```
>>ipod (@)
>>ipad (0)
>>iphone (@)

[K]racked WiFi Scanner v 1 5 3
>>[ (-)
>>k (-)
>>] (-)
>>rack (@)
>>wifi (2) = sip, uma.
>>scanner (10) = printer, peripheral, device, firewall, hardware, ↩
   ↩camera, analyzer, nikon, mouse, monitor.
>>v (-)
>>1 (-)
>>5 (-)
>>3 (-)
```

5.2.5 Discussion

As it was shown above, feature expansion of a short scarce text may cause a *semantic drift* as any new feature may flip the class. For the texts of the positive class, we observed many miscorrections ($x \to y, x \in TP, y \in FN'$) and a very little number of corrections ($x \to y, x \in FN, y \in TP'$). For the texts of the negative class, we observed many corrections ($x \to y, x \in FP, y \in TN'$) and a little number of miscorrections ($x \to y, x \in TN, y \in FP'$). The better performance of the vocabulary projection on the non-pornographic texts is probably due to the semantic relations used in the experiment. These relations were extracted from a general corpus (*WaCky* + *ukWaC*) and therefore contain only a little number of specific pornographic terms. In general, we observed 4,026 more corrections than miscorrections due to the vocabulary projection: $x \to y, x \in (TP \cup TN), y \in (FN' \cup FP')$.

The two likely reasons of the semantic drift are the following:

1. *Uniform weighting.* The original lemmas have the same weight as the lemmas provided by the vocabulary projection. Furthermore, all projected terms are considered as equally important. However, it may be better to weight higher the original terms and the terms highly related to the given out-of-vocabulary lemma. For instance, an improved vocabulary projection could assign a weight w_i to each lemma c_i in the following way:

$$w_i = \begin{cases} 1 & \text{if } c_i \text{ is the original lemma} \\ \frac{q}{r_i} & \text{if } c_i \text{ is the projected lemma} \end{cases} \tag{5.3}$$

Here $q \in [0; 1]$ is the weight of the first projected lemma and r_i is the rank of the projected lemma, according to a semantic similarity measure.

2. *Uniform number of projections.* The current approach searches among $n = 30$ most semantically related terms and returns up to n projected terms. However, for very short texts, this may lead to a semantic drift as the majority of the features would come from the projection, e. g.:

```
Preston My Naughty Massage Copy
>>chanel (@)
>>preston (@)
>>my (-)
>>naughty (@)
>>massage (@)
>>copy (18) = paste, artwork, headline, document, documentation, ↩
   ↩cut, material, delete, photograph, text, print, postage, ↩
   ↩paper, command, braille, file, correspondence, glossy.
```

Thus, it may be better to search among n most similar terms, but use not more than m first matches, where $n > m$. One solution is to choose m depending on the number of original lemmas.

5.2.6 Summary

We have presented the filename classification module, that makes a part of the iCOP toolkit. Our results confirm the correctness of the chosen methodology for filename categorization as the system achieves accuracy of 91% when trained and tested on independent datasets. At the next step, we are going to use the system for the categorization of different kinds of porn (e.g., “gay” vs “lesbian”) and to distinguish CSA media from other porn data.

5.3 Possible Applications to Text-Based Information Retrieval

Above we described two language processing systems which rely on semantic similarity measures. The measures may be used for some other applications such as short text similarity (Mihalcea et al., 2006; Graillet, 2012), text similarity (Steinberger et al., 2002; Tsatsaronis et al., 2010), word sense disambiguation (Agirre and Rigau, 1996; Patwardhan et al., 2003; Bollegala et al., 2007), community mining (Bollegala et al., 2007), anaphora resolution (Poesio et al., 1997; Munoz and Palomar, 2001; Cimiano et al., 2005) or question answering (Sun et al., 2005). This section describes how automatically extracted semantic relations can improve text-based information retrieval systems.

A thesaurus organizes terms of a certain domain with semantic relations between them (see Figure 5.10 and Section 1.1.2). Thesauri have been used in documentation management

projects for years (e. g., DEWEY [29] and other subject headings presented in Section 1.1). They were even used by libraries and documentation centers long before the computer era. This long tradition has led to the adaption of thesaurus-based techniques by the industry and to the development of international standards [30]. Today, thesauri find their place in specialized information retrieval systems (biomedical, legal, etc.).

energy industry
- **NT1 energy conversion**
  - RT soft energy (6626)
- **NT1 energy technology**
  - RT bioconversion (6411)
  - RT oil technology (6616)
  - RT soft energy (6626)
- **NT2 fuel cell**
- **NT1 energy-generating product**
- **NT1 fuel**
  - RT energy resources (5211)
- **NT2 fossil fuel**
  - RT coal (6611)
  - RT natural gas (6616)
  - RT petroleum (6616)

Figure 5.10: EuroVOC thesaurus: a term with its relations.

Techniques described in this thesis, may be useful for automatic thesaurus construction as they establish relations between terms (Fox et al., 1988; Crouch and Yang, 1992; Grefenstette, 1994; Caraballo, 1999; Curran and Moens, 2002; Chen et al., 2003; Nakayama et al., 2007). According to Jones and Willett (1997) and Hodge (2000), once a semantic resource is available, it can be used in a retrieval system for:

- *Indexing* (Lancaster, 1972; Woods, 1997; Pouliquen et al., 2006). In contrast to a traditional full-text search, the system may index only/additionally terms of a thesaurus. The goal here is to improve search precision by avoiding ambiguous indexing terms. Such indexing can also improve recall if the index is expanded with synonyms.
- *Query expansion* (Hodge, 2000; Hsu et al., 2006). Search recall may improve if queries are augmented with synonyms. However, incorrect expansions can hamper precision (Voorhees, 1994; Manning et al., 2008). To keep a high precision, a ranking algorithm should incorporate information about query expansion.
- *Query suggestion* (Catarci et al., 2004; Cao et al., 2008). The goal of query suggestion is to recommend queries related to the initial request. Semantic relations can help to find such related terms. Figure 5.11 illustrates query suggestion of the Yahoo! search engine.

[29] `http://www.oclc.org/dewey/`

[30] The most recent standard (2005) is ANSI/NISO Z39.19-2005: "Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies". The predecessor of this standard is ISO 5964: "Documentation - Guidelines for the establishment and development of monolingual thesauri" (1986)

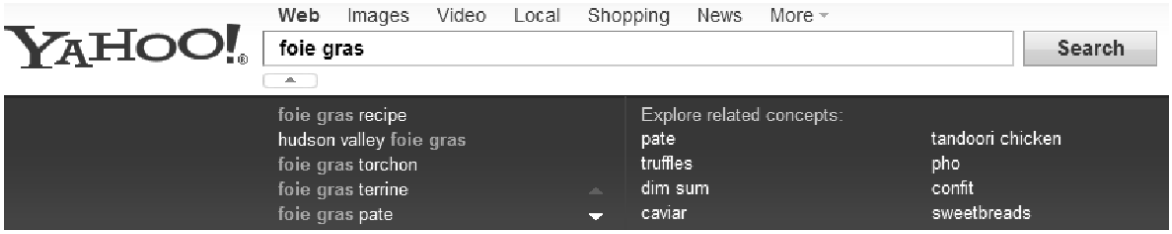


Figure 5.11: Query suggestion function of the Yahoo! search engine.

- *Navigation.* A thesaurus can be used to organize documents hierarchically. For instance, a thesaurus may let a user navigate in the collection from broader to narrower queries (see Figure 5.12).

Nowadays, thesauri are mostly used in specialized systems such as legislative, medical or agricultural search engines. For instance, information retrieval systems CADIAL [31] operates on a collection of Croatian legislative documents. The system relies on the Eurovoc thesaurus in order to implement query expansion, navigation and for document categorization (see Figure 5.12). The system maintains a full-text index and an index of thesaurus descriptors.

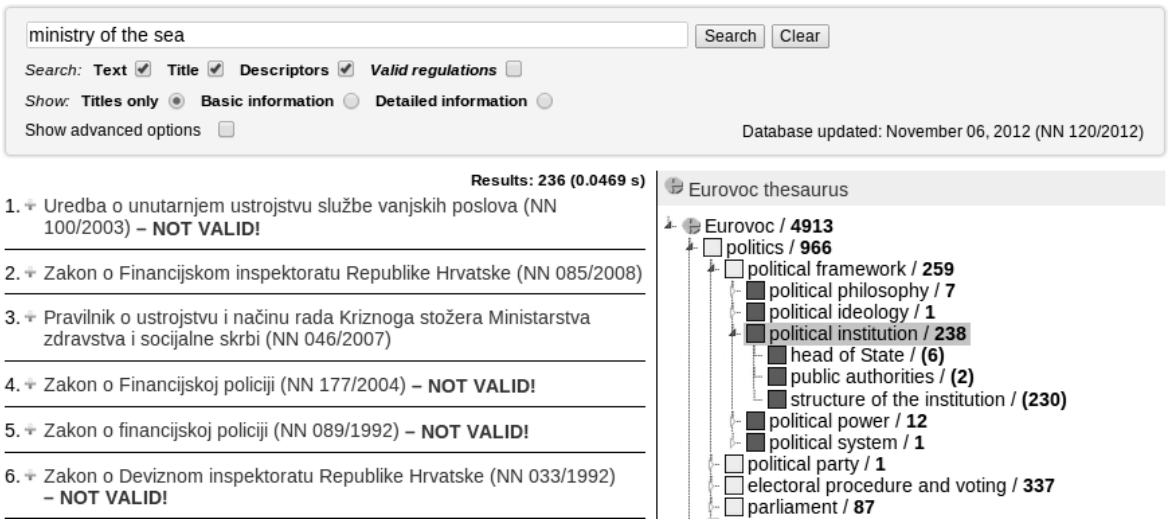


Figure 5.12: Use of the Eurovoc thesaurus in the CADIAL legislative information retrieval system.

PubMed [32] provides access to publications in Life Sciences, Biology and Medicine (see Figure 5.13 (a)). PubMed uses MeSH thesaurus [33] to implement query suggestion and query expansion. Similarly to CADIAL, the system maintains both full-text and thesaurus indexes.

AGRIS [34] is a big online database of agricultural literature. It uses the multilingual thesaurus Agrovoc for indexing, navigation, query expansion, query suggestion and cross-

[31] Computer Aided Document Indexing for Accessing Legislation, `http://www.cadial.org/`
[32] US National Library of Medicine, `http://www.ncbi.nlm.nih.gov/pubmed`
[33] Medical Subject Headings, `http://www.nlm.nih.gov/mesh/`
[34] Information System for the Agricultural Science and Technology, `http://agris.fao.org/`

lingual query expansion (see Figure 5.13 (b)).

Thesauri are often used in digital libraries. For instance, University Information System RUSSIA [35] is an electronic library in Economics, Sociology, Political Science, International Relations and other humanities (see Figure 5.13 (c)). The system uses Socio-Political Thesaurus (Ageev et al., 2006) for indexing, query suggestion and query expansion [36].

5.4 Conclusion

In this chapter, we have presented two language processing applications which use semantic similarity measures. A similarity measure plays a key role in the lexico-semantic search engine "Serelex". It enables the system to retrieve terms semantically related to the query and rank them by relevance. In the second application, a similarity measure is used to improve accuracy of a baseline statistical text classifier. Here, we applied the *vocabulary projection* technique, which substitutes out-of-vocabulary terms with related in-vocabulary terms. Finally, we have provided examples of other text processing applications which may benefit from the methods developed in this work. We conclude that the semantic similarity measures proposed in this thesis may be useful in a wide range of NLP and IR applications.

[35] UIS RUSSIA, `http://uisrussia.msu.ru/`

[36] `http://uisrussia.msu.ru/docs/ips/n/techno/index.htm`

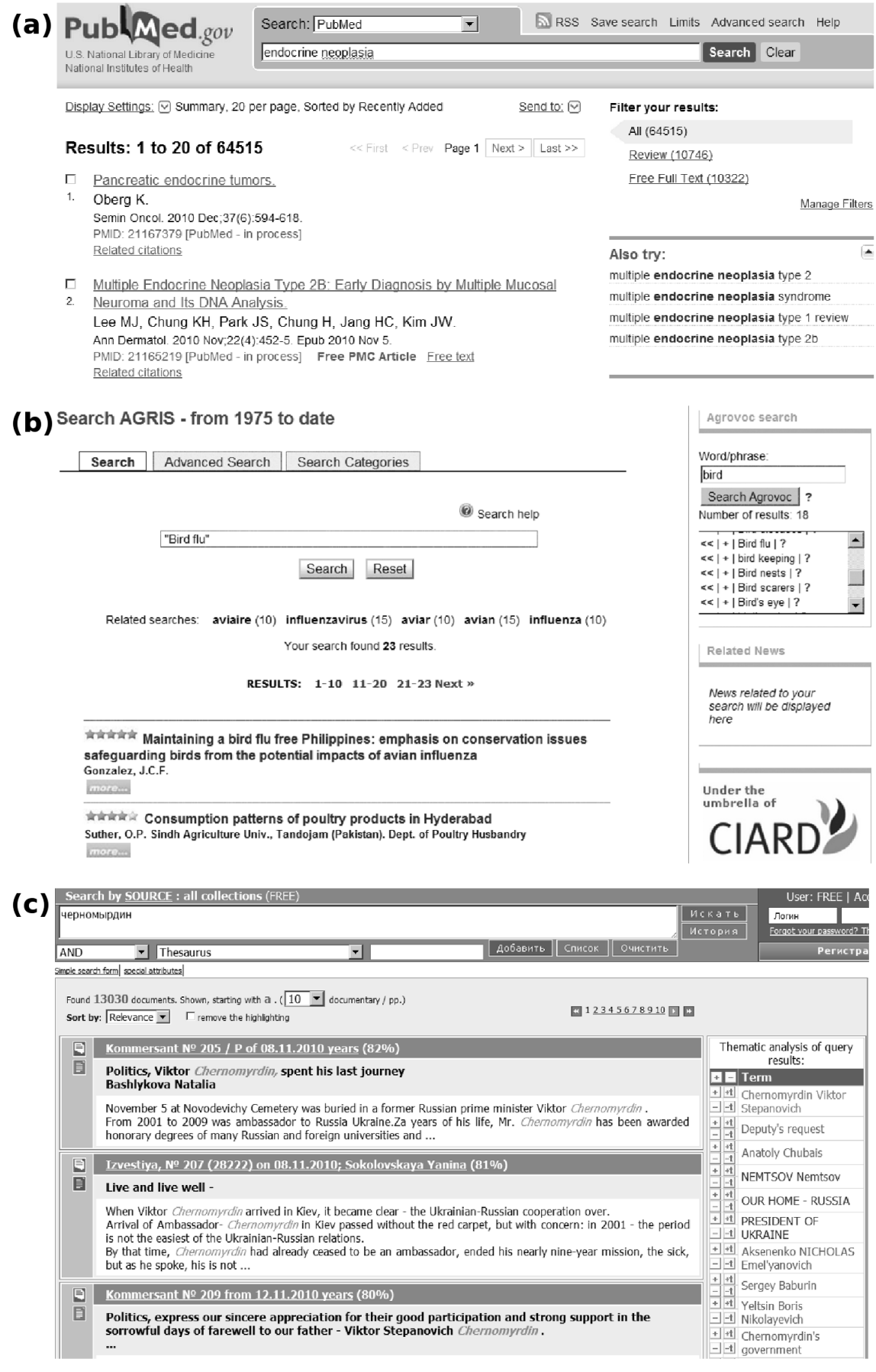


Figure 5.13: (a) use of the MeSH thesaurus in the PubMed medical information retrieval system; (b) use of the AGROVOC thesaurus in the AGRIS information system; (c) use of the Socio-Political Thesaurus in the digital library UIS RUSSIA.

Conclusion

This dissertation investigated several strategies for semantic relation extraction based on similarity measures. Such measures are designed to quantify semantic relatedness of lexical units, such as nouns, noun phrases and multiword expressions. These measures assign high scores to pairs of terms in a semantic relation (synonyms, hypernyms or co-hyponyms) and near-zero values to all other pairs.

The work has brought six key contributions to the field of computational lexical semantics:

1. Section 2.2 of Chapter 1 presented a new similarity measure *SDA-MWE* based on the syntactic distributional analysis and p-NN procedure. The measure performs comparably to the baselines. In contrast to the common approaches, it can deal with both single words and multiword expressions. We compared relations extracted with this measure with relations of a thesaurus. While 7% of the extracted relations is explicitly encoded in the thesaurus, 35%-46% are implicitly present in the resource via the short paths.

2. Section 2.3 of Chapter 1 presented a method for semantic relation extraction *DefVectors*. It relies on Wiktionary, Wikipedia, k-NN, mk-NN procedures and the Vector Space Model. The method performs comparably to the baselines. In contrast to the corpus-based techniques, it operates on a small-scale set of definitions. The proposed technique is implemented in an open source system. [37]

3. Section 2.4 of Chapter 1 described a novel corpus-based semantic similarity measure *PatternSim*, which makes use of the lexico-syntactic patterns. The measure performs comparably to the baseline measures. In contrast to the network-based techniques, it requires no semantic resources such as WordNet or dictionaries. Implementation of the approach has been made available to the community. [38]

[37] https://github.com/jgc128/defvectors
[38] https://github.com/cental/patternsim

4. Chapter 3, presented a large-scale comparative study of 37 baseline similarity measures based on corpora, definitions and semantic networks. We go further than most of the surveys and compare the measures with respect to the semantic relation types they provide. The main findings of this study are the following. The *BDA-3-Cos* and the *SDA-21-Cos* measures provides the best performance among corpus-based measures. The *Resnik* measure performs best among the network-based measures. The *DefVectors-WktWiki* scores best among the definition-based measures. The studied measures are heterogeneous in terms of their lexical coverage, performances and semantic relation distributions. There is no single measure that outperforms all others on all benchmarks. While the semantic relation distributions of the studied measures differ, all these measures extract many co-hyponyms. Evaluation system used in these experiments has been made open source. [39]

5. Chapter 4 provided two main contributions: First, a systematic analysis of 16 baseline measures combined with 9 fusion methods. We are first to propose hybrid similarity measures based on all main types of resources – text corpora, Web as a corpus, semantic networks, dictionaries and encyclopedias. Second, the hybrid supervised semantic similarity measures *Logit-E15*, *C-SVM-radial-E15* and *C-SVM-linear-E15*. They combine 15 baseline measures in a statistical model trained on a set of semantic relations from a dictionary. These measures outperform by a large margin both baseline and unsupervised hybrid measures on all the benchmarks. The key advantages of these measures with respect to the single measures are higher precision (better top results) and recall (better lexical coverage).

6. Chapter 5 presented two text processing systems, which use the semantic similarity measure *PatternSim*. The first system lets users discover semantically similar words in an easy and interactive way. [40] [41] [42] For a given input query, it returns a list of related terms and visualizes them with a graph or a set of images. The system would not function without a similarity measure. Implementation of the system has been made available to the community. [43] [44]

   The second system performs categorization of filenames from the P2P networks to detect child sexual abuse materials. In contrast to the first application, in this case, the similarity measure refines the baseline system. The extracted relations improve the

[39] https://github.com/alexanderpanchenko/sim-eval
[40] http://serelex.cental.be,
[41] http://apps.microsoft.com/webpdp/en-US/app/lsse/48dc239a-e116-4234-87fd-ac90f030d72c,
[42] http://www.windowsphone.com/s?appid=dbc7d458-a3da-42bf-8da1-de49915e0318
[43] https://github.com/pomanob/lsse,
[44] https://github.com/jgc128/serelex4win

accuracy of the application with help of the *vocabulary projection* technique. Implementation of the system is open source. [45]

The two main limitations of this thesis are the following:

1. We deal only with the data in English with the exception of Section 2.2 that presents experiments with the French data.

2. The measures discussed in this work did not perform word sense disambiguation (Lesk, 1986; Agirre and Rigau, 1996). They take as input a term and return a set of its related terms. For instance, if the input term is "python", then the results would contain both terms related to snakes and programming languages. A measure which takes into account word senses, should take into account also context of the term. For instance, if the term "python" occurs in a context related to programming languages (such as words "program" and "java"), then the results should contain only terms related to programming.

Three practical questions regarding the semantic similarity measures are the following:

- "Should a given text processing application use a semantic similarity measure?"
- "How to integrate a semantic similarity measure into a given application?"
- "Which semantic similarity measure should be used in a given application?"

This thesis mainly deals with the last question. Our general advice is to use the hybrid supervised similarity measures, such as *C-SVM-radial-E15* wherever possible. However, they require various linguistic resources: corpora, dictionaries, free access to Web search engines, semantic networks, training data, etc. If all these components are available for the vocabulary of your application, we advice to use such advanced measures. However, in some domains, such resources does not exist. In this case, first you should collect as much resources as possible covering the target vocabulary. Next, you should use the measures or a combination of measures, which match the available linguistic resources. For instance, if you have access to a text corpus and a dictionary then you should test a combination of corpus- and dictionary-based measures.

Therefore, this thesis improved understanding of the existing approaches to semantic similarity and proposed several new ones. These novel techniques perform well according to both extrinsic and intrinsic evaluations. We conclude that the developed measures can be useful in a wide range of natural language processing and information retrieval applications.

[45] `https://github.com/alexanderpanchenko/stc`

Finally, this thesis has identified four prominent directions for the future work:

1. Applying the proposed measures to the NLP systems dealing with short texts, such as filenames, sentences, abstracts, short messages, tags, tweets or Facebook statuses. Examples of such systems include text categorization systems, text retrieval systems, systems measuring text similarity, machine translation systems, text clustering systems, etc. It is often desirable to enrich representation of the short texts (compare *query expansion* or *vocabulary projection*). One way to do it is to use only synonyms and other semantically related terms.

2. Development of the relation-specific similarity measures. Such measures would assign high scores only to hypernyms or hyponyms or synonyms, etc. For certain applications we may prefer relations of a specific type. For instance, for the *query expansion* we would prefer to use synonyms. One way to implement such technology is to use the supervised similarity measures proposed in Chapter 4. These measures should be trained on a relation-specific data. Some additional relation-specific features may be also needed.

3. Development of the semantic similarity measures for other languages. Multiple measures exist for English. However, such tools are not available for many other languages. Porting the measures to other languages may improve NLP applications of those languages. Some measures described in this thesis can be straightforwardly applied to other languages, e. g. *BDA, SDA, LSA* and *DefVectors*. The *PatternSim* measure can be also ported to another language. In this case, a reasonable effort is needed to translate the extraction patterns.

4. Supporting multiword expressions (MWEs) and named entities (NEs). This work mostly focused on the similarity measures dealing with single nouns, with exception of the technique described in Section 2.2. However, processing of MWEs and NEs is important for many text processing systems. Therefore, it would be interesting to extend the proposed approaches so they fully support MWEs and NEs.

Bibliography

Ageev, M., Dobrov, B., and Lukashevich, N. (2008). Automatic text categorization: Methods and problems. *Kazanskii Gosudarstvennyi Universitet. Uchenye Zapiski*, 150(4):25–40.

Ageev, M., Dobrov, V., and Loukachevitch, N. (2006). Socio-Political Thesaurus in Concept-Based Information Retrieval. *Accessing Multilingual Information Repositories*, pages 141–150.

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., and Soroa, A. (2009). A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of NAACL-HLT 2009*, pages 19–27.

Agirre, E. and Rigau, G. (1996). Word sense disambiguation using conceptual density. In *Proceedings of the International Conference on Computational Linguistics (COLING'96).*, pages 16–22. Association for Computational Linguistics.

Agresti, A. (2002). *Categorical Data Analysis (Wiley Series in Probability and Statistics)*. 2 edition.

Aït-Mokhtar, S., Chanod, J., and Roux, C. (2002). Robustness beyond shallowness: incremental deep parsing. *Natural Language Engineering*, 8(3):121–144.

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Asuka, S., Naoki, Y., and Kentaro, T. (2008). Boosting precision and recall of hyponymy relation acquisition from hierarchical layouts in wikipedia. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Auger, A. and Barrière, C. (2008). Pattern-based approaches to semantic relation extraction: A state-of-the-art. *Terminology Journal*, 14(1):1–19.

Avriel, M. (2003). *Nonlinear programming: analysis and methods*. Dover Publications.

Baeza-Yates, R. and Tiberi, A. (2007). Extracting semantic relations from query logs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 76–85. ACM.

Balkova, V., Suhonogov, A., and Yablonsky, S. (2004). Rusian WordNet: From UML-notation to Internet/Infranet Database Implementation. In *Proceedings of the Second International WordNet Conference (GWC 2004)*.

Banerjee, S. and Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *IJCAI*, volume 18, pages 805–810.

Barnes, J. and Hut, P. (1986). A hierarchical 0 (n log iv) force-calculation algorithm. *nature*, 324:4.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *LREC*, 43(3):209–226.

Baroni, M. and Lenci, A. (2011). How we blessed distributional semantic evaluation. *GEMS (EMNLP), 2011*, pages 1–11.

Bentivogli, L., Pianta, E., and Girardi, C. (2002). Multiwordnet: developing an aligned multilingual database. In *First International Conference on Global WordNet, Mysore, India*.

Bernard, J. R. (1990). *The Macquarie Encyclopedic Thesaurus.* The Macquarie Library, Sydney, Australia.

Bingham, E. and Mannila, H. (2001). Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM.

Bishop, C. et al. (2006). *Pattern recognition and machine learning*, volume 4. springer New York.

Blei, D., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Blondel, V., Guillaume, J., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Blondel, V. and Senellart, P. (2002). Automatic extraction of synonyms in a dictionary. *vertex*, 1:x1.

Bogdanova, D., Petersburg, S., Rosso, P., and Solorio, T. (2012). On the impact of sentiment and emotion based features in detecting online sexual predators. *WASSA 2012*, pages 110–118.

Bollegala, D., Matsuo, Y., and Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. In *WWW*, volume 766.

Budanitsky, A. and Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

Budiu, R., Royer, C., and Pirolli, P. (2007). Modeling information scent: A comparison of lsa, pmi and glsa similarity measures on common tests and corpora. pages 314–332. In RIAO.

Bullinaria, J. A. and Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.

Candito, M., Nivre, J., Denis, P., and Anguiano, E. (2010). Benchmarking of statistical dependency parsers for french. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 108–116. Association for Computational Linguistics.

Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E., and Li, H. (2008). Context-aware query suggestion by mining click-through and session data. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 875–883. ACM.

Caraballo, S. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 120–126. Association for Computational Linguistics.

Caraballo, S. A. (2001). *Automatic construction of a hypernym-labeled noun hierarchy from text*. PhD thesis, Providence, RI, USA. Adviser-Charniak, Eugene.

Catarci, T., Dongilli, P., Di Mascio, T., Franconi, E., Santucci, G., and Tessaris, S. (2004). An ontology based visual tool for query formulation support. In *ECAI 2004: 16th European Conference on Artificial Intelligence, August 22-27, 2004, Valencia, Spain: including Prestigious Applicants [sic] of Intelligent Systems (PAIS 2004): proceedings*, page 308. Ios Pr Inc.

Cederberg, S. and Widdows, D. (2003). Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings HLT-NAACL*, page 111–118.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chen, L. (2006). *Automatic construction of domain-specific concept structures*. PhD thesis, Universitats-und Landesbibliothek Darmstadt.

Chen, Z., Liu, S., Wenyin, L., Pu, G., and Ma, W. (2003). Building a web thesaurus from web link structure. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 48–55. ACM.

Cilibrasi, R. L. and Vitanyi, P. M. B. (2007). The Google Similarity Distance. *IEEE Trans. on Knowl. and Data Eng.*, 19(3):370–383.

Cimiano, P. (2006). *Ontology learning and population from text: algorithms, evaluation and applications*. Springer Verlag.

Cimiano, P., Saric, J., and Reyle, U. (2005). Ontology-driven discourse analysis for information extraction. *Data and Knowledge Engineering*, 55(1):59–83.

Cohen, T., Schvaneveldt, R., and Widdows, D. (2010). Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, 43(2):240–256.

Corral, M. (2008). *Vector Calculus*. Michael Corral.

Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Crouch, C. (1988). A cluster-based approach to thesaurus construction. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 309–320. ACM.

Crouch, C. J. and Yang, B. (1992). Experiments in automatic statistical thesaurus construction. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 77–88, New York, NY, USA. ACM Press.

Curran, J. R. (2002). Ensemble methods for automatic thesaurus extraction. In *Proceedings of the EMNLP-02*, pages 222–229. ACL.

Curran, J. R. (2003). *From distributional to semantic similarity*. PhD thesis, University of Edinburgh.

Curran, J. R. and Moens, M. (2002). Improvements in automatic thesaurus extraction. In *Proceedings of the ACL-02 workshop on Unsupervised Lexical Acquisition*, pages 59–66.

de Melo, G. and Weikum, G. (2009). Towards a universal wordnet by learning from combined evidence. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 513–522. ACM.

Dhillon, I., Guan, Y., and Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM.

Dice, L. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

Dobrov, V. (2002). Automatic categorization of full-text documents with complex classification schemes. (in russian: Avtomaticheskaya rubrikatsiya polnotekstovih dokumentov po klassifikatoram slojnoi strukturi.). In *8th Conference on Artificial Intelligence (CIA-02). Russian Conference open for international participation.*, pages 7–12.

Ellman, J. (2003). Eurowordnet: A multilingual database with lexical semantic networks. *Natural Language Engineering*, 9(04):427–430.

Erk, K. and Padó, S. (2008). A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 897–906. Association for Computational Linguistics.

Fan, R., Chang, K., Hsieh, C., Wang, X., and Lin, C. (2008). Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Fellbaum, C. (1998). *WordNet: An electronic lexical database*. The MIT press.

Ferret, O. (2010). Testing semantic similarity measures for extracting synonyms from a corpus. In *Proceeding of LREC*.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: The concept revisited. In *WWW 2001*, pages 406–414.

Fleiss, J. (1971). Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Floyd, R. W. (1962). Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75–174.

Fox, E., Nutter, J., Ahlswede, T., Evens, M., and Markowitz, J. (1988). Building a large thesaurus for information retrieval. In *Proceedings of the second conference on Applied natural language processing*, pages 101–108. Association for Computational Linguistics.

Fruchterman, T. M. J. and Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 6, page 12.

Gangemi, A., Navigli, R., and Velardi, P. (2003). The OntoWordNet Project: extension and axiomatization of conceptual relations in WordNet. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, pages 820–838.

Graillet, O. S. (2012). Article: Semantic similarity measure for pairs of short biological texts. *International Journal of Applied Information Systems*, 4(5):1–5. Published by Foundation of Computer Science, New York, USA.

Grefenstette, G. (1993). Automatic thesaurus generation from raw text using knowledge-poor techniques. In *In Making Sense of Words. Ninth Annual Conference of the UW Centre for the New OED and text Research*.

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery (The Springer International Series in Engineering and Computer Science)*. Springer.

Griffiths, T., Steyvers, M., et al. (2003). Prediction and semantic association. *Advances in neural information processing systems*, pages 11–18.

Griffiths, T., Steyvers, M., and Tenenbaum, J. (2007). Topics in semantic representation. *Psychological review*, 114(2):211.

Gurevych, I. (2005). Using the structure of a conceptual network in computing semantic relatedness. *Natural Language Processing–IJCNLP 2005*, pages 767–778.

Hadj Taieb, M., Ben Aouicha, M., Tmar, M., and Ben Hamadou, A. (2012). Wikipedia category graph and new intrinsic information content metric for word semantic relatedness

measuring. In Xiang, Y., Pathan, M., Tao, X., and Wang, H., editors, *Data and Knowledge Engineering*, Lecture Notes in Computer Science, pages 128–140. Springer Berlin Heidelberg.

Hall, D., Jurafsky, D., and Manning, C. (2008). Studying the history of ideas using topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 363–371. Association for Computational Linguistics.

Hall, J., Nilsson, J., and Nivre, J. (2011). Single malt or blended? a study in multilingual parser optimization. volume 43 of *Text, Speech and Language Technology*, pages 19–33. Springer.

Hamp, B. and Feldweg, H. (1997). Germanet-a lexical-semantic net for german. In *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15. Citeseer.

Hanks, P. (2000). *The New Oxford Thesaurus of English.* Oxford University Press, Oxford, UK.

Harris, Z. (1954). Distributional structure. *Word*.

Hatzivassiloglou, V. and McKeown, K. (1993). Towards the automatic identification of adjectival scales: Clustering adjectives according to meaning. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 172–182. Association for Computational Linguistics.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *ACL*, pages 539–545.

Heylen, K., Peirsman, Y., Geeraerts, D., and Speelman, D. (2008). Modelling word similarity: an evaluation of automatic synonymy extraction algorithms. *LREC'08*, pages 3243–3249.

Hickok, R. (1995). *Roget's II: the new thesaurus. Third edition.* Boston, MA USA.

Hindle, D. (1990). Noun classification from predicate-argument structures. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, pages 268–275. Association for Computational Linguistics.

Hirschman, L., Grishman, R., and Sager, N. (1975). Grammatically-based automatic word class formation. *Information Processing & Management*, 11(1-2):39–57.

Ho, N. D. and Fairon, C. (2004). Lexical similarity based on quantity of information exchanged-synonym extraction. *Proceedings of the Research Informatics Vietnam-Francophony, Hanoi, Vietnam*, pages 193–198.

Hodge, G. (2000). *Systems of knowledge organization for digital libraries*. The Digital Library Federatio. Concil on Library and Information Resources. Washington, DC.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196.

Hosmer, W. and Stanley, L. (2000). Applied logistic regression. *John Wiley&amp*.

Howell, D. C. (2010). *Fundamental statistics for the behavioral sciences*. Wadsworth Pub Co.

Hsu, M.-H., Tsai, M.-F., and Chen, H.-H. (2006). Query expansion with conceptnet and wordnet: An intrinsic comparison. *Information Retrieval Technology*, pages 1–13.

Hughes, T. and Ramage, D. (2007). Lexical semantic relatedness with random graph walks. In *Proceedings of EMNLP-CoNLL*, pages 581–589.

Jaccard, P. (1901). *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz.

Jiang, J. J. and Conrath, D. W. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *ROCLING X*, pages 19–33.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, pages 137–142.

Jones, K. and Galliers, J. (1995). *Evaluating natural language processing systems: An analysis and review*, volume 1083. Springer Verlag.

Jones, K. and Willett, P. (1997). *Readings in information retrieval*. Morgan Kaufmann Pub.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.

Kennedy, A. and Szpakowicz, S. (2008). Evaluating rogets thesauri. *ACL-08 HLT*, pages 416–424.

Kestemont, M., Peersman, C., Decker, D., Pauw, D., Luyckx, K., Morante, R., Vaassen, F., Loo, V., Daelemans, W., et al. (2012). The netlog corpus: a resource for the study of flemish dutch internet language. *LREC'2012*.

Kevers, L. (2009). Indexation semi-automatique de textes: thésaurus et transducteurs. In *Actes de CORIA09 (Sixième Conférence Francophone en Recherche d'Information et Applications)*, pages 151–167.

Kevers, L., Mantrach, A., Fairon, C., Bersini, H., and Saerens, M. (2011). Classification supervisée hybride par motifs lexicaux étendus et classificateurs SVM. *Actes Des 10e Journées Internationales D'analyse Des Données Textuelles, Rome (June 2010)*.

Kilgarriff, A. and Grefenstette, G. (2003). Introduction to the special issue on the web as corpus. *Computational linguistics*, 29(3):333–347.

Kittler, J. (1998). Combining classifiers: A theoretical framework. *Pattern Analysis & Applications*, 1(1):18–27.

Kobozeva, I. (2009). *Component Analysis of Lexical Meaning (Komponentnii analiz leksicheskogo znacheniya)*. In: Linguistical Semantics, 4rd ed., "LIBRICOM", Moscow, Russia.

Kontostathis, A., West, W., Garron, A., Reynolds, K., and Edwards, L. (2012). Identifying predators using chatcoder 2.0. *PAN-2012*.

Kullback, S. and Leibler, R. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Kuncheva, L. (2007). Combining pattern classifiers: Methods and algorithms (kuncheva, li; 2004)[book review]. *Neural Networks, IEEE Transactions on*, 18(3):964–964.

Lancaster, F. (1972). *Vocabulary control for information retrieval.* Information Resources Press, 2100 M Street, NW, Washington, DC 20037.

Landauer, T. K. and Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psych. review*, 104(2):211.

Leacock, C. and Chodorow, M. (1998). Combining Local Context and WordNet Similarity for Word Sense Identification. *An Electronic Lexical Database*, pages 265–283.

Lee, L. (1999). Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 25–32. Association for Computational Linguistics.

Lenci, A. and Benotto, G. (2012). Identifying hypernyms in distributional semantic spaces. pages 75–79.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM.

Lin, D. (1998a). An Information-Theoretic Definition of Similarity. In *ICML*, pages 296–304.

Lin, D. (1998b). Automatic retrieval and clustering of similar words. In *ACL*, pages 768–774.

Lin, D. and Pantel, P. (2001). Induction of semantic classes from natural language text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 317–322. ACM.

Lindsey, R., Veksler, V. D., Grintsvayg, A., and Gray, W. D. (2007). Be wary of what your computer reads: the effects of corpus selection on measuring semantic relatedness. In *8th International Conference of Cognitive Modeling, ICCM*.

Liu, H. and Singh, P. (2004). Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Loukachevitch, N., Ageev, M., and Dobrov, V. (2002). Text categorization tasks for large hierarchical systems of categories. *SIGIR 2002 Workshop on Operational Text Classification Systems, F.Sebastiani, S.Dumas, D.D.Lewis, T.Montgomery, I.Moulinier - Univ. of Tampere*, pages 49–52.

Lu, Y., Mei, Q., and Zhai, C. (2011). Investigating task performance of probabilistic topic models: an empirical study of plsa and lda. *Information Retrieval*, 14(2):178–203.

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods*, 28(2):203–208.

Mahalanobis, P. (1936). On the generalized distance in statistics. In *Proceedings of the National Institute of Sciences of India*, volume 2, pages 49–55. New Delhi.

Manning, C., Raghavan, P., and Sch
"utze, H. (2008). *An introduction to information retrieval*. Cambridge University Press.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, 1 edition.

Martin, T. and Azmi-Murad, M. (2005). An incremental algorithm to find asymmetric word similarities for fuzzy text mining. *Soft Computing as Transdisciplinary Science and Technology*, pages 838–847.

Matveeva, I. (2007). Term representation with generalized latent semantic analysis irina matveeva*, gina-anne levow*, ayman farahat" & christiaan royer*** university of chicago,** palo alto research center. *Recent advances in natural language processing IV: selected papers from RANLP 2005*, 292:45.

McGhee, I., Bayzick, J., Kontostathis, A., Edwards, L., McBride, A., and Jakubowski, E. (2011). Learning to identify internet sexual predation. *International Journal of Electronic Commerce*, 15(3):103–122.

McRae, K., Cree, G., Seidenberg, M., and McNorgan, C. (2005). Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547–559.

Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI'06*, pages 775–780.

Miller, G. (1995a). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Miller, G. A. (1995b). Wordnet: a lexical database for english. *Communications of ACM*, 38(11):39–41.

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Miller, G. A., Leacock, C., Tengi, R., and Bunker, R. T. (1993). A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. ACL.

Milne, D., Medelyan, O., and Witten, I. H. (2006). Mining domain-specific thesauri from wikipedia: A case study. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 442–448, Washington, DC, USA. IEEE Computer Society.

Moens, M. (2006). *Information extraction: algorithms and prospects in a retrieval context*, volume 1. Springer.

Morlane-Hondère, F. and Fabre, C. (2012). Étude des manifestations de la relation de méronymie dans une ressource distributionnelle (study of meronymy in a distribution-based lexical resource) [in french]. In *Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 2: TALN*, pages 169–182, Grenoble, France. ATALA/AFCP.

Muller, P., Hathout, N., and Gaume, B. (2006). Synonym extraction using a semantic distance on a dictionary. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pages 65–72. Association for Computational Linguistics.

Munoz, R. and Palomar, M. (2001). Semantic-driven algorithm for definite description resolution. In *Recent Advances in Natural Language Processing (RANLP)*, pages 180–186.

Nakayama, K., Hara, T., and Nishio, S. (2007). Wikipedia mining for an association web thesaurus construction. pages 322–334.

Navarro, E., Sajous, F., Bruno, G., Prévot, L., ShuKai, H., Tzu-Yi, K., Magistry, P., and Chu-Ren, H. (2009). Wiktionary and nlp: improving synonymy networks. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, People's Web '09, pages 19–27. Association for Computational Linguistics.

Navigli, R. and Ponzetto, S. P. (2010). BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* Uppsala, Sweden, 11–16 July 2010, pages 216–225.

Nelson, D., McEvoy, C., and Schreiber, T. (2004). The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods*, 36(3):402–407.

Padó, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Panchenko, A. (2011). Comparison of the baseline knowledge-, corpus-, and web-based similarity measures for semantic relations extraction. *GEMS Workshop (EMNLP)*, pages 11–21.

Panchenko, A. (2012). A study of heterogeneous similarity measures for semantic relation extraction. In *Proceedings of 14e Rencontres des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (JEP-TALN-RECITAL 2012)*, pages 29–42.

Panchenko, A. and Morozova, O. (2012). A study of hybrid similarity measures for semantic relation extraction. In *Proceedings of Innovative Hybrid Approaches to the Processing of Textual Data Workshop. EACL 2012*, pages 10–18.

Panchenko, A., Morozova, O., and Naets, H. (2012). A semantic similarity measure based on lexico-syntactic patterns. In Jancsary, J., editor, *Proceedings of KONVENS 2012*, pages 174–178. ÖGAI. Main track: poster presentations.

Pantel, P., Ravichandran, D., and Hovy, E. (2004). Towards terascale knowledge acquisition. In *Proceedings of the 20th international conference on Computational Linguistics*, page 771. Association for Computational Linguistics.

Patwardhan, S., Banerjee, S., and Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, volume 2588 of *LNCS*, pages 241–257. Springer Berlin.

Patwardhan, S. and Pedersen, T. (2006). Using WordNet-based context vectors to estimate the semantic relatedness of concepts. *Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*, page 1.

Paumier, S. (2003). *De la reconnaissance de formes linguistiques à l'analyse syntaxique*. PhD thesis, Université de Marne-la-Vallée.

Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41. ACL.

Peersman, C., Daelemans, W., and Van Vaerenbergh, L. (2011). Predicting age and gender in online social networks. In *Proceedings of the 3rd international workshop on Search and mining user-generated contents*, pages 37–44. ACM.

Peersman, C., Vaassen, F., Van Asch, V., and Daelemans, W. (2012). Conversation level constraints on pedophile detection in chat rooms. *PAN-2012*.

Pendar, N. (2007). Toward spotting the pedophile telling victim from predator in text chats. In *Semantic Computing, 2007. ICSC 2007. International Conference on*, pages 235–241. IEEE.

Pereira, F., Tishby, N., and Lee, L. (1993). Distributional clustering of english words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 183–190. Association for Computational Linguistics.

Philippovich, Y. N. and Prokhorov, A. V. (2002). *Semantics of Information Technologies (In Russian: Semantika Informatsionnikh Technologii: Opiti Slovarno-Tesaurusnogo Opisaniya)*. ISBN 5-8122-0367-9, MGUP, Moscow, Russia.

Poesio, M., Vieira, R., and Teufel, S. (1997). Resolving bridging references in unrestricted text. In *Proceeding of the Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts.*

Poole, D. (2010). *Linear algebra: A modern introduction*. Brooks/Cole Publishing Company.

Pouliquen, B., Steinberger, R., and Ignat, C. (2006). Automatic annotation of multilingual text collections with a conceptual thesaurus. *Arxiv preprint cs/0609059*.

Resnik, P. (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *IJCAI*, volume 1, pages 448–453.

Rogers, D. and Tanimoto, T. (1960). A computer program for classifying plants. *Science*, 132(3434):1115–1118.

Roget, P. (1911). *Roget's Thesaurus of English words and phrases...* TY Crowell co.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Rybinski, H., Kryszkiewicz, M., Protaziuk, G., Jakubowski, A., and Delteil, A. (2007). Discovering synonyms based on frequent termsets. In *RSEISP '07: Proceedings of the international conference on Rough Sets and Intelligent Systems Paradigms*, pages 516–525, Berlin, Heidelberg. Springer-Verlag.

Sagot, B. and Fišer, D. (2008). Building a free french wordnet from multilingual resources. In *Proceedings of OntoLex*. Citeseer.

Sahlgren, M. (2006). *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis.

Sang, E. and Hofmann, K. (2007). Automatic extraction of dutch hypernym-hyponym pairs. *Proceedings of CLIN-2006. Leuven, Belgium*.

Sang, E. and Hofmann, K. (2009). Lexical patterns or dependency patterns: which is better for hypernym extraction? In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 174–182. Association for Computational Linguistics.

Schaeffer, S. (2007). Graph clustering. *Computer Science Review*, 1(1):27–64.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. pages 44–49.

Schölkopf, B., Smola, A., Williamson, R., and Bartlett, P. (2000). New support vector algorithms. *Neural computation*, 12(5):1207–1245.

Schütze, H. (1993). Word space. In *Advances in Neural Information Processing Systems 5*. Citeseer.

Séaghdha, D. (2010). Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 435–444. Association for Computational Linguistics.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Selz, O. (1913). Uber die gesetze des geordneten denkvertaufs. *Stuttgart: Spemann.*

Senellart, P. and Blondel, V. (2008). Automatic discovery of similarwords. *Survey of Text Mining II*, pages 25–44.

Snow, R., Jurafsky, D., and Ng, A. Y. (2004). Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems (NIPS)*, 17:1297–1304.

Sorenson, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species content. *Kongelige Danske Videnskabernes Selskab*, 5(1-34):4–7.

Sowa, J. (1983). Conceptual structures: information processing in mind and machine.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., and Demirbas, M. (2010). Short text classification in twitter to improve information filtering. In *Proceeding of the 33rd international ACM SIGIR conference on research and development in information retrieval*, pages 841–842. ACM.

Stamou, S., Oflazer, K., Pala, K., Christoudoulakis, D., Cristea, D., Tufis, D., Koeva, S., Totkov, G., Dutoit, D., and Grigoriadou, M. (2002). BALKANET A Multilingual Semantic Network for the Balkan Languages. In *Proceedings of the International Wordnet Conference, Mysore, India*, pages 21–25.

Steinberger, R., Pouliquen, B., and Hagman, J. (2002). Cross-lingual document similarity calculation using the multilingual thesaurus eurovoc. *A. Gelbukh (Ed.): CICLing 2002, LNCS. Springler-Verlag Berlin Heidelberg*, 2276:415–424.

Strube, M. and Ponzetto, S. P. (2006). Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the AAAI*, volume 21, pages 14–19.

Sun, R., Jiang, J., Fan, Y., Hang, T., Tat-seng, C., and yen Kan, C. M. (2005). Using syntactic and semantic relation analysis in question answering. In *Proceedings of TREC*.

Takenobu, T., Makoto, I., and Hozumi, T. (1995). Automatic thesaurus construction based on grammatical relations. In Mellish, C., editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1308–1313, San Francisco. Morgan Kaufmann.

Tan, Pang-Ning Steinbach, M. K. V. (2006). *Introduction to data mining*. Pearson.

Tax, D., Van Breukelen, M., Duin, R., and Kittler, J. (2000). Combining multiple classifiers by averaging or by multiplying? *Pattern recognition*, 33(9):1475–1485.

Theodoridis, S. and Koutroumbas, K. (2009). *Pattern Recognition*. Elsevier.

Tikk, D., Yang, J. D., and Bang, S. L. (2003). Hierarchical text categorization using fuzzy relational thesaurus. *KYBERNETIKA-PRAHA*, 39(5):583–600.

Tsatsaronis, G., Varlamis, I., and M., V. (2010). Text relatedness based on a word thesaurus. *Journal of Artificial Intelligence Research*, 37:1–39.

Tudhope, D., Koch, T., and Heery, R. (2006). Terminology Services and Technology: JISC State of the art review. *Retrieved January*, 8:2007.

Turney, P. (2001). Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the ECML-2001*.

Ulges, A. and Stahl, A. (2011). Automatic detection of child pornography using color visual words. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6. IEEE.

Van Assem, M., Gangemi, A., and Schreiber, G. (2006). Conversion of WordNet to a standard RDF/OWL representation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06), Genoa, Italy*. Citeseer.

Van de Cruys, T. (2010). *Mining for Meaning: The Extraction of Lexicosemantic Knowledge from Text*. PhD thesis, University of Groningen.

Van Der Plas, L. and Bouma, G. (2004). Syntactic contexts for finding semantically related words. In *Computational linguistics in the Netherlands*.

Vapnik, V. (1999). *The nature of statistical learning theory*. springer.

Veksler, V. D., Govostes, R. Z., and Gray, W. D. (2008). Defining the dimensions of the human semantic space. In *30th Annual Meeting of the Cognitive Science Society*, pages 1282–1287.

Voorhees, E. (1994). Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69. Springer-Verlag New York, Inc.

Wandmacher, T. (2005). How semantic is Latent Semantic Analysis? *Proceedings of TALN/RECITAL*.

Ward, G. (1996). *Moby Thesaurus.* Moby Lexicon Project.

Wellisch, H. (1991). *Indexing from A to Z.* HW Wilson.

Woods, W. (1997). Conceptual indexing: A better way to organize knowledge. *Sun Microsystems, Inc. Mountain View, CA, USA*.

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of ACL'1994*, pages 133–138.

Xu, L., Krzyzak, A., and Suen, C. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(3):418–435.

Yang, H. and Callan, J. (2009). A metric-based framework for automatic taxonomy induction. In *ACL-IJCNLP*, page 271–279.

Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM.

Yeh, E., Ramage, D., Manning, C., Agirre, E., and Soroa, A. (2009). Wikiwalk: random walks on wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 41–49. Association for Computational Linguistics.

Yen, L., Fouss, F., Decaestecker, C., Francq, P., and Saerens, M. (2007). Graph nodes clustering based on the commute-time kernel. *Advances in Knowledge Discovery and Data Mining*, pages 1037–1045.

Zesch, T. and Gurevych, I. (2007). Analysis of the wikipedia category graph for nlp applications. In *Proceedings of the TextGraphs-2 Workshop (NAACL-HLT 2007)*, pages 1–8.

Zesch, T., Gurevych, I., and Mühlhäuser, M. (2007). Comparing wikipedia and german wordnet by evaluating semantic relatedness on multiple datasets. In *HLT-NAACL 2007*, pages 205–208.

Zesch, T., Muller, C., and Gurevych, I. (2008a). Extracting lexical semantic knowledge from wikipedia and wiktionary. In *Proceedings of LREC'08*, pages 1646–1652.

Zesch, T., Muller, C., and Gurevych, I. (2008b). Using wiktionary for computing semantic relatedness. In *Proceedings of AAAI*, page 45.

Appendix A: Additional Examples of the Serelex System

This appendix contains examples of the *Serelex* system described in Section 5.1:

- Figures 14, 15 and 16 provide additional examples of the web-interface based on the images from Google Image Search. This web interface is described in Section 5.1.
- Figure 17 (a) demonstrates the graphical user interface of the Serelex desktop client. It is available for download from the Microsoft Windows Store and requires Microsoft Windows 8 or Windows RT [46]. The source code of this application is available under conditions of the LGPLv3 license [47].
- Figure 17 (b) illustrates the graphical user interface of the Serelex application for Microsoft Windows Phone 8. This free application is available for download from the Microsoft Windows Store [48].

[46] http://apps.microsoft.com/webpdp/en-US/app/lsse/48dc239a-e116-4234-87fd-ac90f030d72c.

[47] https://github.com/jgc128/serelex4win

[48] http://www.windowsphone.com/s?appid=dbc7d458-a3da-42bf-8da1-de49915e0318

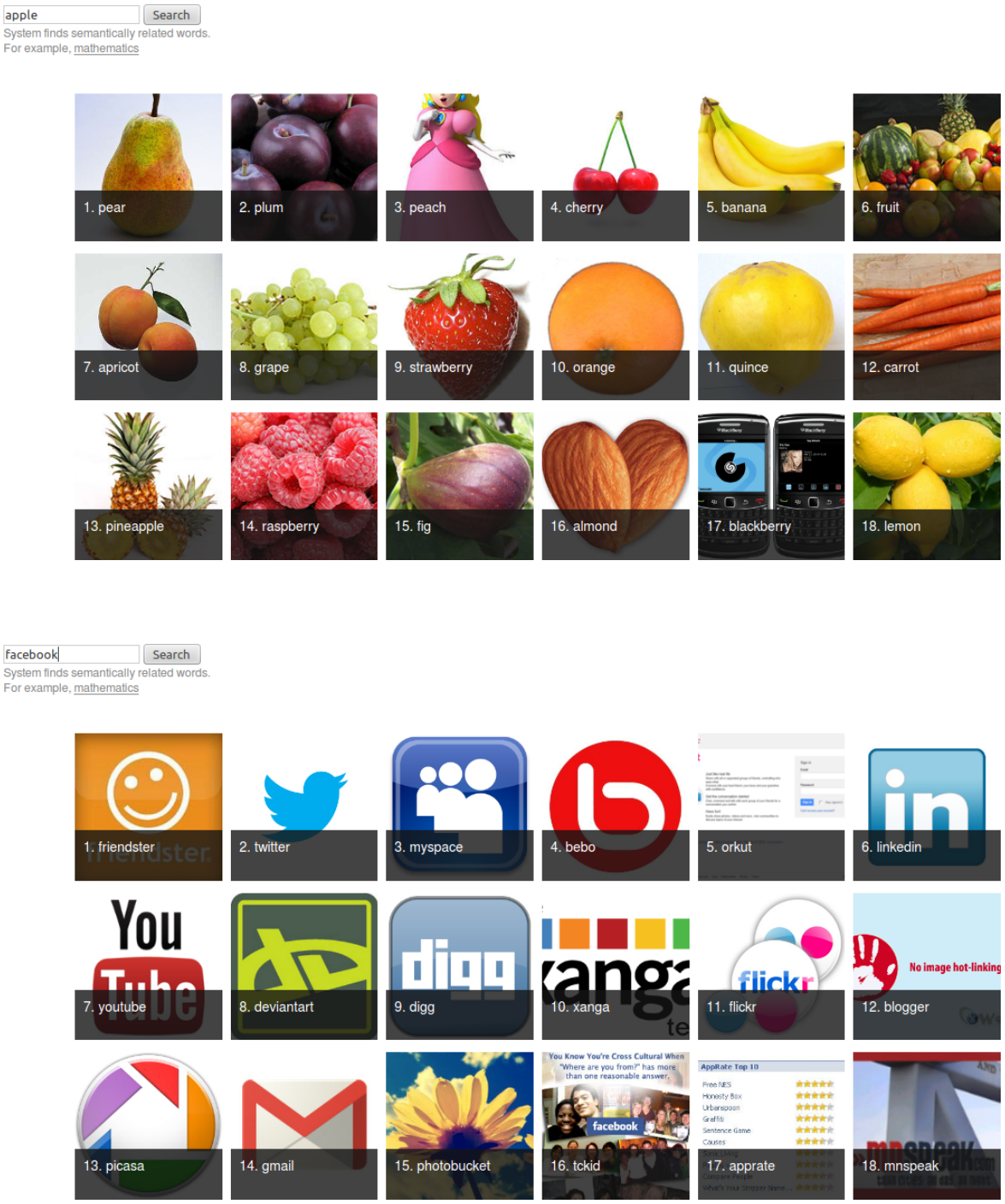


Figure 14: Graphical user interface of the "Serelex" system: queries "apple" and "Facebook".

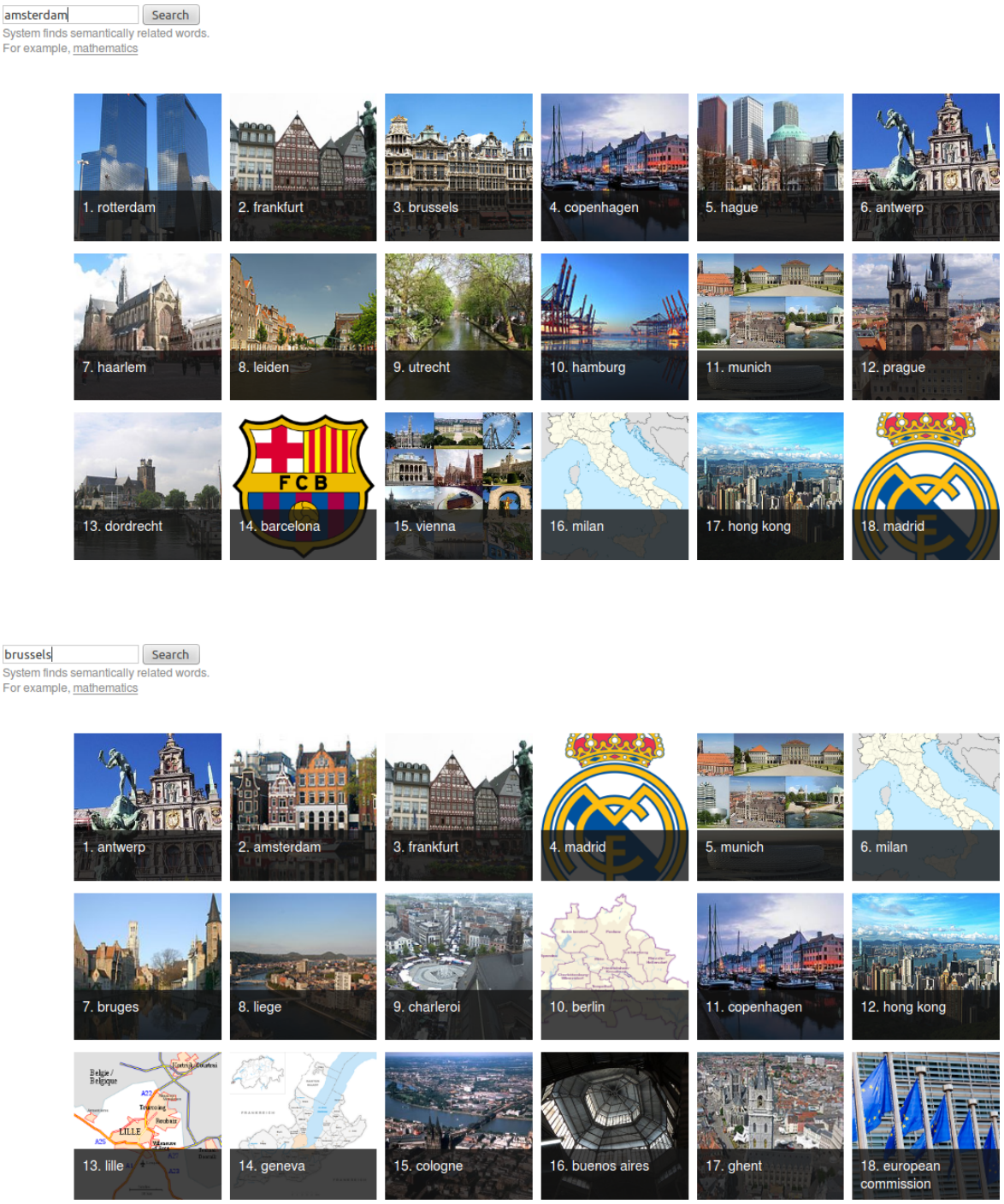


Figure 15: Graphical user interface of the "Serelex" system: queries "Amsterdam" and "Brussels".

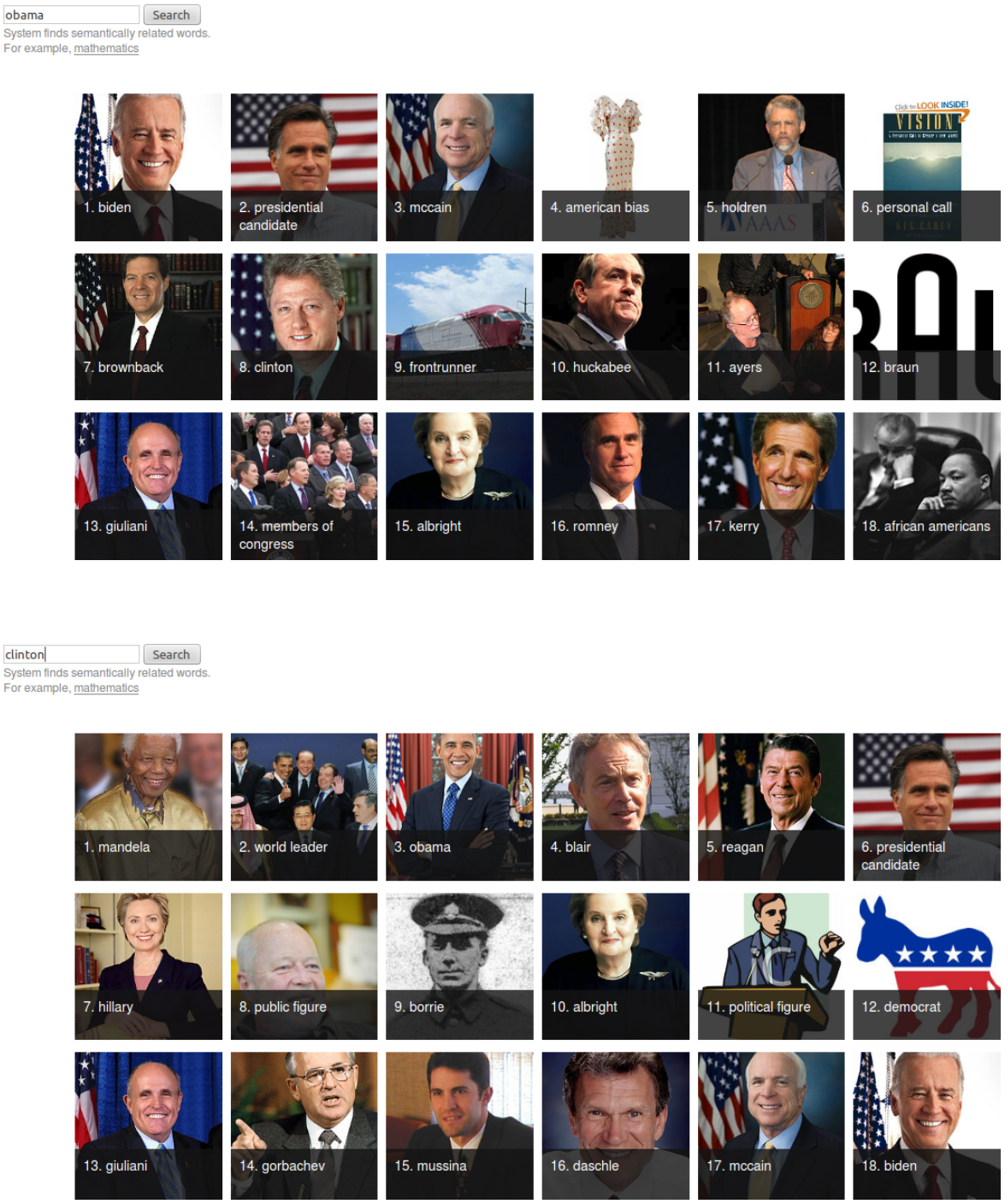
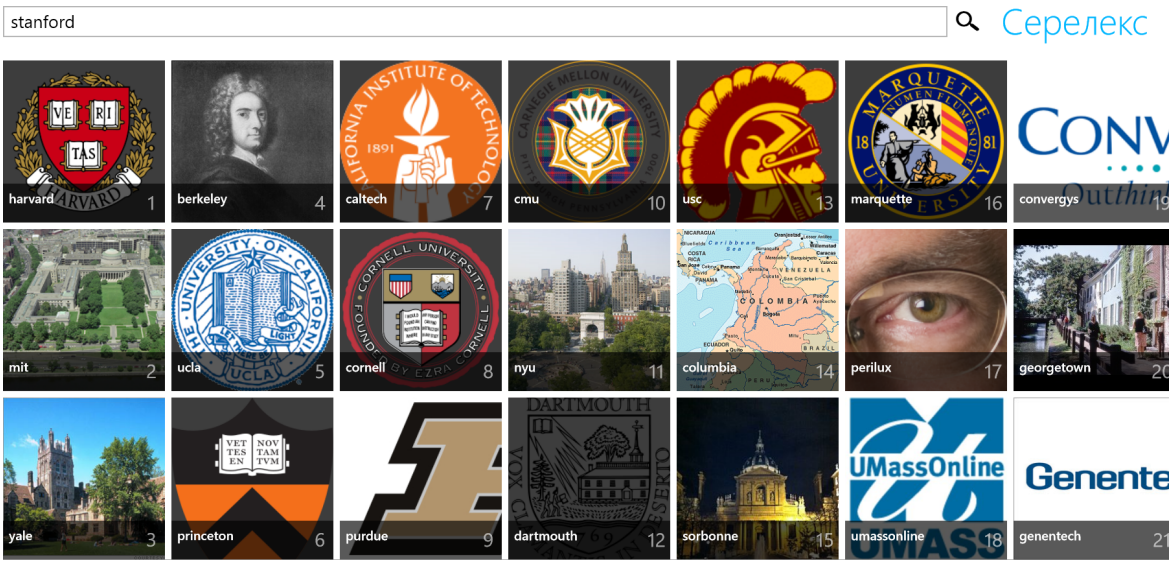
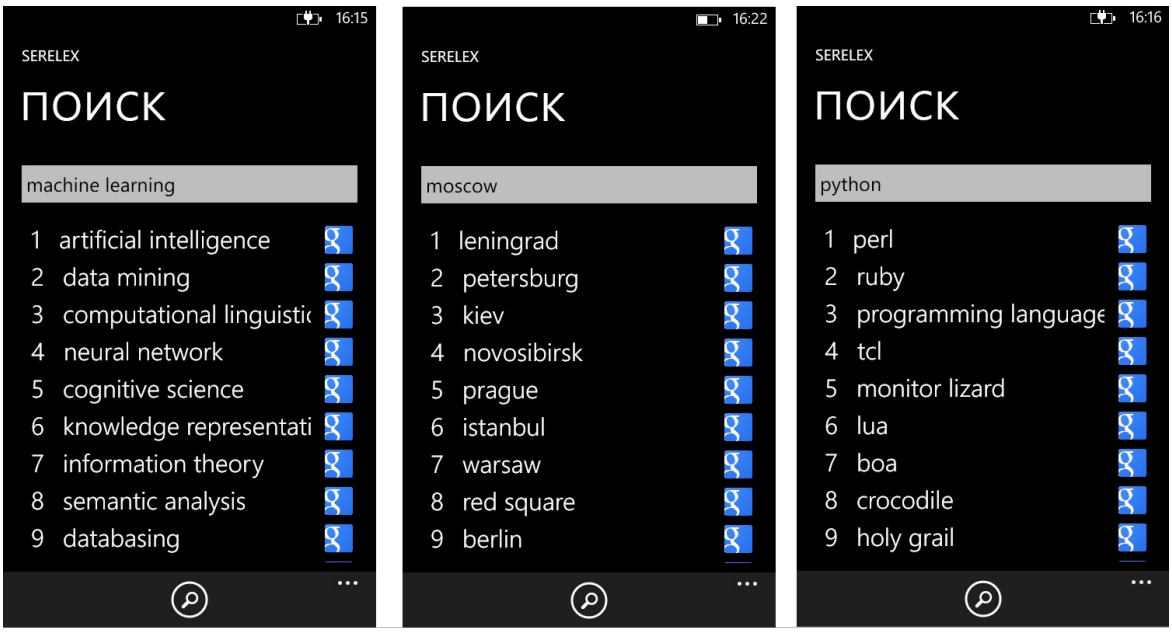


Figure 16: Graphical user interface of the "Serelex" system: queries "Obama" and "Clinton".



(a)



(b)

Figure 17: (a) Serelex client for Windows 8: query "Stanford"; (b) Serelex client for Windows Phone 8: queries "machine learning", "Moscow" and "python".