ENRICO MENSA

USING SENSES MAKES SENSE: BUILDING LEXICAL RESOURCES FOR SENSE INFORMED NLP APPLICATIONS

University of Turin

Polo delle Scienze della Natura

Computer Science Department

Doctorial Thesis

USING SENSES MAKES SENSE: BUILDING LEXICAL RESOURCES FOR SENSE INFORMED NLP APPLICATIONS

ENRICO MENSA

A language is not just words. It's a culture, a tradition, a unification of a community, a whole history that creates what a community is. It's all embodied in a language.

— Noam Chomsky

In all affairs it's a healthy thing now and then to hang a question mark on the things you have long taken for granted.

— Bertrand Russell

ABSTRACT

Lexical resources are fundamental to tackle many tasks that are central to present and prospective research in Text Mining, Information Retrieval and, of course, to Natural Language Processing at large. In this scope, semantic lexical resources have been proven to be particularly useful to develop successful applications in various areas. In this work two lexical resources are presented: COVER and LessLex. COVER proposes a compact vectorial representation that combines the lexicographic precision characterizing BabelNet and the rich common-sense knowledge featuring ConceptNet, while LessLex is a set of embeddings that extends the terminological embeddings of ConceptNet Numberbatch by building semantic representations that co-exist in the same semantic space with those acquired at the term level: for each term we have thus the 'blended' terminological vector along with those describing all senses associated to that term. Such conceptual representations are language independent; as illustrated in the experimentation, this enables to deal with multilingual and cross-lingual settings. These resources have been extensively tested on a wide variety of tasks, such as word similarity, conceptual similarity, semantic similarity with explanation, keyword extraction, abstractness computation, metaphor detection, contextual similarity and semantic text similarity. The obtained results seem to corroborate the main hypothesis underlying this work: sense-level representations —as opposed to term-level representations— typically require additional processing efforts (such

as, e.g., for some sort of disambiguation), but favorably compare to term-based representations, that have been providing state-of-the-art results in the last few years, at the same time allowing to produce more cognitively plausible and explainable results.

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to the prof. Daniele Radicioni, that guided and followed me during this years, definitively proving that having the right supervisor and person by your side makes all the difference. A very special thanks goes also to my colleague Davide Colla, for his tireless dedication and presence.

I'm extremely grateful to my family, for their unparalleled support that allowed me to pursue this career.

I also wish to thank Ilenia, Carlo, Fabio and all of my closest friends for their patience and immense care.

CONTENTS

III APPENDIX

LIST OF FIGURES

LIST OF TABLES

Part I

BASICS

1

INTRODUCTION

Lexical resources are fundamental to tackle many tasks that are central to present and prospective research in Natural Language Processing: in the last decades, the success in several tasks such as word sense disambiguation has been strongly related to the development of lexical resources (Miller, 1995; Miller and Fellbaum, 2007; Navigli, 2009). The same holds for specialized forms of semantic analysis and interpretation, such as sentiment analysis, where systems' efficacy (Cambria, Schuller, et al., 2013) has been accompanied by the release of specialized lexical resources and corpora (e.g., (Bosco, Patti, and Bolioli, 2013; Devitt and Ahmad, 2013; McCrae et al., 2012)). In the last few years the creation of multilingual and parallel resources (Francopoulo et al., 2009; Navigli and Ponzetto, 2010) further strengthened the link between lexical resources and successful NLP applications (Denecke, 2008; Gînscă et al., 2011; Moro, Cecconi, and Navigli, 2014), while the impact of deep architectures and word embeddings has been compared to a tsunami hitting the NLP Community and its major conferences (Manning, 2015).

Word embeddings have been successfully applied to a broad —and still growing— set of diverse application fields, such as computing the similarity between short texts (Kenter and De Rijke, 2015), full documents (Kusner et al., 2015) or both (Le and Mikolov, 2014). Also by looking at traditional NLP tasks such as parsing, embeddings proved to be an effective instrument for syntactical parsing —both dependency (Bansal, Gimpel, and Livescu, 2014; Hisamoto, Duh, and Matsumoto, 2013) and constituency parsing (Andreas and Klein, 2014)— and semantic parsing as well (Berant and P. Liang, 2014). Within this phenomenon, multilingual and cross-lingual word embeddings have gained a special status, thanks to the strong and partly unanswered pressure for devising tools and systems to deal with more than one language at a time. Amongst the main areas where multilingual and cross-lingual resources and approaches are solicited there are of course machine translation (Cho et al., 2014; Luong, Pham, and Manning, 2015), cross-lingual document categorization (Gouws, Bengio, and Corrado, 2015) and sentiment analysis (Tang et al., 2014).

The focus of this work is the development of two lexical resources which constitute the main contributions of my PhD course. The first resource, COVER (so named after 'COmmon-sense VEctorial Representation'), is a set of conceptual common-sense vectors that were proposed as a helpful resource to semantically elaborate text documents. COVER is built by merging BabelNet (Navigli and Ponzetto, 2012), NASARI (Camacho-Collados, Pilehvar, and Navigli, 2015b) and ConceptNet (Havasi, Robyn Speer, and Alonso, 2007) with the aim at combining, in a synthetic and cognitively grounded way, lexicographic precision and common-sense aspects. Different from most popular vectorial resources that rely on Distributional Semantics, representing hundreds of opaque distributional features, COVER provides the represented elements with a reduced number of cognitively salient dimensions and, as illustrated in the following, it allows building applications that obtain interesting results in a number of tasks.

The second proposed resource is LessLex, a novel set of embeddings containing descriptions for *senses* rather than for terms. Word embeddings typically describe terms (with few notable exceptions such as NASARI (Camacho-Collados, Pilehvar, and Navigli, 2015b) and SenseEmbed (Iacobacci, Pilehvar, and Navigli, 2015)). This means that different (though close) vectorial descriptions are collected for terms such as *table*, *board*, *desk* for each considered language; whilst in a resource based on senses just one description for the sense of *table* (e.g., intended as “a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs”) would suffice. Of course this fact has consequences on the number of vectors involved in multilingual and cross-language applications: one vector per term per language in the case of terminological vectors, one per sense —regardless of the language— otherwise.

However, since one major challenge in the lexical semantics field is, to date, that of dealing with as many as possible languages at the same time (e.g., BabelNet covers 284 different languages to date),[1] we decided to explore the hypothesis that to deal with multilingual applications, and even more in cross-lingual ones, systems can benefit from compact, concept-based representations. The final result is a semantic resource that enriches the terminological space offered by ConceptNet Numberbatch (Robyn Speer, Chin, and Havasi, 2017) by also generating distributional vectors for each term meaning, borrowed from the BabelNet sense inventory. The evaluation of our vectors seems to support our hypotheses: LessLex vectors have been tested in a widely varied experimental setting, providing performances at least on par with

[1] https://babelnet.org/stats.

state-of-the-art embeddings, and sometimes substantially improving on these.

LessLex and COVER grasp different and complementary types of knowledge: the former supplies distributional and opaque knowledge which can be exploited to establish distances both on a terminological and conceptual level. The latter encodes conceptual and human readable knowledge which could be used to further refine semantic distance computation, while also providing explainable insights on the adopted computational processes. This complementarity is enabled by a *shared semantic layer* between the two resources: concepts' indexing is performed in both cases based on the BabelNet semantic inventory, thus making the two resources tightly connected and the knowledge therein truly interoperable.

2

STATE OF THE ART

In the last few years different methodologies and systems for the construction of unified lexical and semantic resources have been proposed, as illustrated in Figure 2.1. The timeline shows a clear shift: until a decade ago, most of the resources could be arranged into hand-crafted resources — created either by expert annotators, such as WordNet (Miller, 1995), FrameNet (Baker, Fillmore, and Lowe, 1998) and VerbNet (Levin, 1993), or through collaborative initiatives, such as ConceptNet (Havasi, Robyn Speer, and Alonso, 2007) —; and resources that have been built by automatically combining the above ones, like in the case of BabelNet (Navigli and Ponzetto, 2012). These resources feature the advantage of being human readable and therefore the behavior of the applications that are built on top of them is often more interpretable. However, in the last decade these explanatory traits have been disregarded in favor of the high performance and usability ensured by the usage of *word embeddings*. Since the foundational work that gave birth to word2vec (Mikolov, Sutskever, et al., 2013), this new line of research has brought huge improvements in almost every NLP task. Further works allowed the development of other types of embeddings, such as multilingual embeddings (e.g., ConceptNet Numberbatch (Robyn Speer, Chin, and Havasi, 2017)), sense embeddings (e.g., SenseEmbed (Iacobacci, Pilehvar, and Navigli, 2015)) and finally contextual embeddings (e.g., ELMo (Peters et al., 2018)).

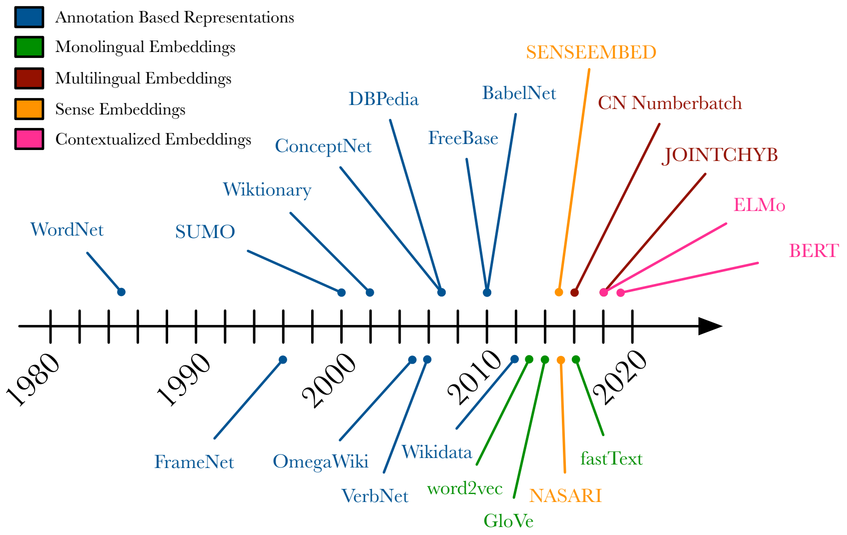


Figure 2.1: Mapping on the timeline of some of the most relevant linguistic resources proposed in the last decades.

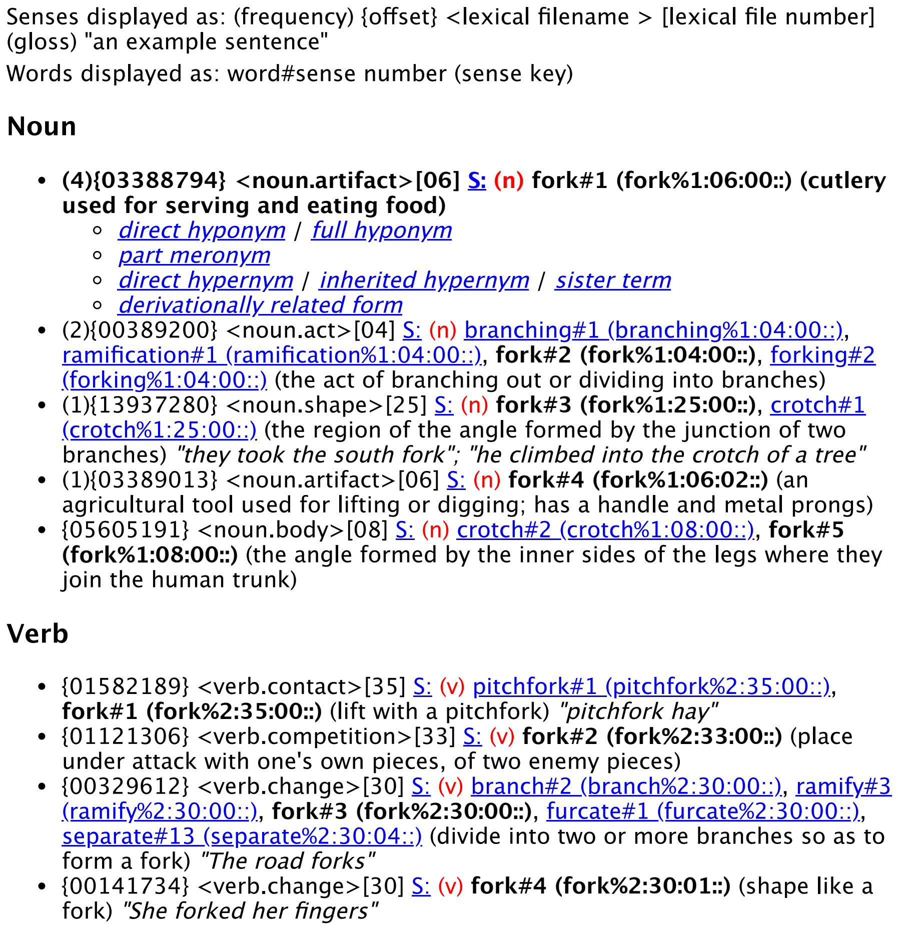
Our two contributions constitute a complementing effort to enrich this scene: from one side COVER preserves the explainable capabilities of the earlier resources while focusing on the common-sense knowledge which has been rarely considered in literature. On the other side, LessLex joins the embeddings front by providing multilingual conceptual vectors. Both resources share the trait of being *conceptually grounded* to the BabelNet inventory and thus allow for the resolution of high level NLP tasks.

In the following sections we will provide a description of some of the most relevant resources in this line of research, focusing especially on those that we relied upon in order to build COVER and LessLex.

2.1 SEMANTIC NETWORKS

Semantic networks are knowledge bases in which nodes (*synsets*) represent uniquely identified *concepts*. These nodes are linked via various semantic relationships and additionally enriched with information about the represented concept such as its lexicalizations, a gloss, etc.. Semantic networks constitute the backbone of any semantic-aware application since they provide the sense inventory (i.e., the “sense vocabulary”) around which applications are developed. WordNet and BabelNet constitute two of the most adopted and relevant networks developed in the field.

WORDNET. WordNet (Miller, 1995) constitutes the first attempt at building a lexical database for English. It is manually curated and its version 3.0 contains around 120.000 synsets and more than 150.000 unique terms. Each node (synset) represents a set of synonyms, expressing a distinct sense. These nodes are also endowed with a gloss and various examples of use of the described concept, provided for differential purposes, that is for distinguishing senses. WordNet is actually partitioned in four categories, modeled upon the four *open-class* parts of speech: nouns, verbs, adjectives and adverbs. Each portion of WordNet has its own relations connecting entities herein. Nouns are organized in a lexical memory as hierarchies, verbs are organized by a variety of entailment relations, while adjectives and adverbs are organized as N-dimensional hyperspaces: each of these lexical structures reflects a different way of categorizing experience. Relations linking nouns senses are not only hyponymy and hypernymy (which are at the base of the hierarchy), but also include antonymy, meronymy/wholonymy. Verbal relations include troponymy and entailment. As an example, Figure 2.2 shows the list of synsets associated with the term *fork* in WordNet 3.1. Since its publication in 1995 it has been considered as the most



Senses displayed as: (frequency) {offset} <lexical filename > [lexical file number] (gloss) "an example sentence"

Words displayed as: word#sense number (sense key)

Noun

- **(4){03388794} <noun.artifact>[06] S: (n) fork#1 (fork%1:06:00::) (cutlery used for serving and eating food)**
  - *direct hyponym* / *full hyponym*
  - *part meronym*
  - *direct hypernym* / *inherited hypernym* / *sister term*
  - *derivationally related form*
- (2){00389200} <noun.act>[04] S: (n) branching#1 (branching%1:04:00::), ramification#1 (ramification%1:04:00::), **fork#2 (fork%1:04:00::)**, forking#2 (forking%1:04:00::) (the act of branching out or dividing into branches)
- (1){13937280} <noun.shape>[25] S: (n) **fork#3 (fork%1:25:00::)**, crotch#1 (crotch%1:25:00::) (the region of the angle formed by the junction of two branches) *"they took the south fork"; "he climbed into the crotch of a tree"*
- (1){03389013} <noun.artifact>[06] S: (n) **fork#4 (fork%1:06:02::)** (an agricultural tool used for lifting or digging; has a handle and metal prongs)
- {05605191} <noun.body>[08] S: (n) crotch#2 (crotch%1:08:00::), **fork#5 (fork%1:08:00::)** (the angle formed by the inner sides of the legs where they join the human trunk)

Verb

- {01582189} <verb.contact>[35] S: (v) pitchfork#1 (pitchfork%2:35:00::), **fork#1 (fork%2:35:00::)** (lift with a pitchfork) *"pitchfork hay"*
- {01121306} <verb.competition>[33] S: (v) **fork#2 (fork%2:33:00::)** (place under attack with one's own pieces, of two enemy pieces)
- {00329612} <verb.change>[30] S: (v) branch#2 (branch%2:30:00::), ramify#3 (ramify%2:30:00::), **fork#3 (fork%2:30:00::)**, furcate#1 (furcate%2:30:00::), separate#13 (separate%2:30:04::) (divide into two or more branches so as to form a fork) *"The road forks"*
- {00141734} <verb.change>[30] S: (v) **fork#4 (fork%2:30:01::)** (shape like a fork) *"She forked her fingers"*

Figure 2.2: All the synsets associated with the term *fork* in WordNet. The relationships stemming from the first sense are also shown.

reliable sense inventory in literature; it has then been translated in other languages (Pianta, Bentivogli, and Girardi, 2002) and versions suited to the resolution of different tasks such as sentiment analysis have been proposed, as well (Baccianella, Esuli, and Sebastiani, 2010).

BABELNET. BabelNet (Navigli and Ponzetto, 2012) is a multilingual lexicalized semantic network and ontology, containing almost 16 million synsets and about 800 million word senses (distributed over more than 284 languages). Its basic structure is borrowed from WordNet since BabelNet is built by automatically linking Wikipedia pages to WordNet synsets. BabelNet also distinguishes from WordNet in virtue of its multilingual traits: in fact, each node contains multilingual lexicalizations that are collected by exploiting the *inter-language* links of Wikipedia together with a machine translation system. Specifically, the generation of BabelNet can be divided in three steps:

1. WordNet and Wikipedia are combined by automatically acquiring a mapping between WordNet senses and Wikipages. This mapping algorithm relies on the conditional probability of a WordNet sense given a Wikipedia page and is based on the disambiguation contexts generated from the two resources. A proper mapping is required in order to avoid sense duplication and to obtain a sense inventory containing complement of the sense inventories of WordNet and Wikipedia.

2. Multilingual lexicalizations are obtained by using *i*) human-generated translations (established via inter-language links in Wikipedia) and *ii*) a machine translation system.

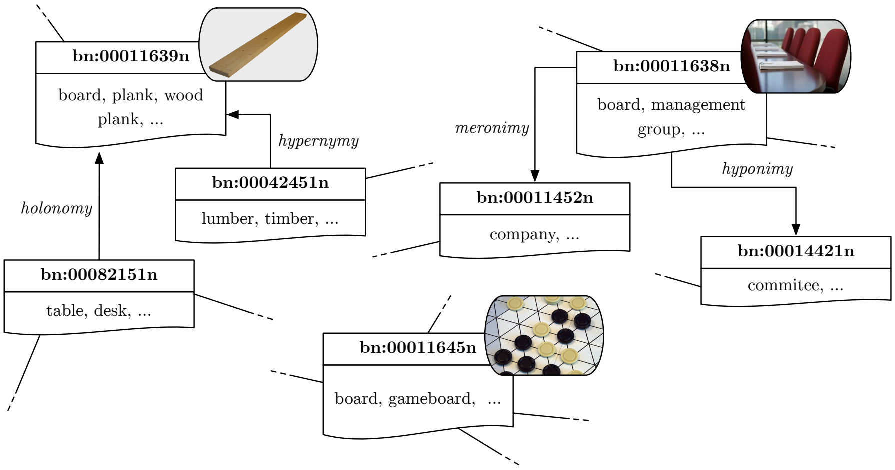


Figure 2.3: An excerpt of the BabelNet network (only English lexicalizations are shown).

3. Relationships between synsets are inherited from WordNet and further expanded by considering the degree of correlation between the two Wikipedia pages associated to the nodes.

The final resource consists in a semantic network in which nodes (called Babel synsets) offer multilingual lexicalizations and are linked by all the WordNet relationships plus an underspecified relatedness relation inherited by the Wikipedia page links.

Further works have been focused on injecting in BabelNet other information extracted from other resources such as Wikidata (Vrandečić and Krötzsch, 2014), ImageNet (Deng et al., 2009), Open Multilingual Wordnet,[1] OmegaWiki[2] and Wikitionary.[3] Figure 2.3 shows an excerpt of the network.

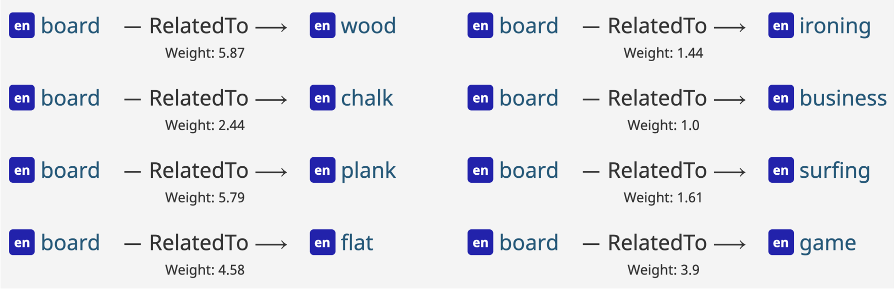
CONCEPTNET. ConceptNet (H. Liu and Singh, 2004) is a semantic network where nodes represent words and compound concepts (e.g., *buy food*, *drive car*) connected through a rich set of pragmatic relation-

[1] http://compling.hss.ntu.edu.sg/omw/.
[2] http://www.omegawiki.org.
[3] https://www.wiktionary.org/.

ships (e.g., *EffectOf*, *AtLocation*). It is focused on the representation of *common-sense* knowledge, which is very hard to be scraped or elicited. Authors motivate the need of a resource such as ConceptNet with the fact that "*common-sense knowledge spans a huge portion of human experience, but is typically omitted from social communications*". Differently from WordNet and Cyc[4] –the two most relevant attempts at knowledge representation available when ConceptNet was conceived– ConceptNet is not handcrafted by knowledge engineers but instead it is automatically generated from the English sentences of the Open Mind Common Sense (OMCS) corpus. The OMCS project, launched in 2000 at the MIT Media Lab, is a crowdsourcing project that allowed more than 14.000 Web contributors to enter sentences in a fill-in-the-blank fashion (e.g., 'A fork is used for ...', 'A table is usually located at ...'). By applying natural language processing and extraction rules to the 700.000 semistructured OMCS sentences, the authors were able to build a network containing more then 300.000 nodes connected via 1.6 million binary-relation assertions. Another peculiar trait of ConceptNet consists in the variety of encoded relationships since they are not only limited to the WordNet-like classical semantic relationships, but also extended to other pragmatic relationships such as *EffectOf* (causality), *SubeventOf* (event hierarchy), *CapableOf* (agent's ability), *PropertyOf*, *LocationOf*, and *MotivationOf* (affect).

Finally, the knowledge in ConceptNet is more practical, defeasible and informal compared to that available in WordNet. For instance, thanks to WordNet we can retrieve the fact that *dog is-a canine* which is in turn a *carnivore*, but we cannot retrieve the more practical member-to-

[4] Cyc is a handcrafted ontology containing over 1.6 million facts interrelating more than 118.000 concepts (http://sw.opencyc.org/).



en board	— RelatedTo ⟶ Weight: 5.87	en wood	en board	— RelatedTo ⟶ Weight: 1.44	en ironing
en board	— RelatedTo ⟶ Weight: 2.44	en chalk	en board	— RelatedTo ⟶ Weight: 1.0	en business
en board	— RelatedTo ⟶ Weight: 5.79	en plank	en board	— RelatedTo ⟶ Weight: 1.61	en surfing
en board	— RelatedTo ⟶ Weight: 4.58	en flat	en board	— RelatedTo ⟶ Weight: 3.9	en game

Figure 2.4: Some of the relationships associated to the node *board* in ConceptNet.

set-association *dog is-a pet*. Moreover, in ConceptNet we can find that *EffectOf*('fall off bicycle', 'get hurt') which is not necessarily always true and therefore cannot be found in WordNet. Subsequent releases of the resource allowed for the integration of information taken from DBPedia (Auer et al., 2007), Wikitionary, and OpenCyc.

It is however relevant to point out that ConceptNet does not provide a semantic layer to its nodes. By looking at some of the relationships attached to the node *board* (Figure 2.4), we can in fact notice that all of its meanings (board as a plank of wood, as a game board, as a management board, as a surfing board and as and ironing board) are conflated into a single node, thus mixing alla such senses. The injection of such semantic layer to govern nodes and their associated concepts could however solve this issue and improve the accessibility of the information in ConceptNet.

2.2 WORD AND SENSE EMBEDDINGS

Word and Sense embeddings are collections of vectors that represent words or senses via a list of N real numbers. Such vectors co-exist in a N-dimension hyperspace in which the geometrical distance among them can be computed, allowing for the calculation of the similarity between words or senses. In this setting, one major assumption is that words that occur in similar contexts tend to purport similar meanings (Harris, 1954); this principle seems to be compatible with some mechanisms of language acquisition that are based on similarity judgments (Yarlett and Ramscar, 2008). Word embeddings have gained enormous success in the community due to their simplicity of use, versatility and great performances in a plethora of NLP tasks. Furthermore, they can also be trained for specific domains.

2.2.1 *Monolingual Word Embeddings*

The development of word2vec (Mikolov, Sutskever, et al., 2013) can be seen as the first successful attempt at building effective word embeddings. The word2vec models and the associated off the shelf word embeddings result from a training over 100 billion words from the Google News through continuous skip-grams. The authors of this work exploit simple — compared to either feedforward or recurrent network models — model architectures and illustrate how to train high quality word vectors from huge data sets. Another resource worth mentioning is GloVe: while word2vec is commonly acknowledged to be a *predictive* model, GloVe (Pennington, Socher, and Manning, 2014) is instead a *count based* model (more on this distinction can be found in the work by Baroni, Dinu, and Kruszewski (2014)). In count based models, model vectors are learned by applying dimensionality reduction techniques to the co-occurrence counts matrix; in particular, GloVe embeddings have been acquired through a training on 840 billion words from the

Common Crawl dataset.[5] The work by Faruqui, Dodge, et al. (2014) has also shown how *retrofitting* can be used to improve vectors quality. Retrofitting is a post-processing step that updates vectors by running a belief-propagation algorithm on a graph constructed from lexicon-derived relational information. This technique is also at the core of ConceptNet Numberbatch (more on this resource in Section 2.1).

Finally, one of the latest contribution in this field is fastText (Bojanowski et al., 2016), which exploits a skipgram model where each word is represented as a bag of character *n*-grams. Compared to other standard models that assign distinct vectors to each word, fastText focuses on morphology, that allows the resource to deal with out-of-vocabulary (OOV) words by also making the training process much faster. Moreover, fastText vectors are also able to capture hidden information about a language, like word analogies or semantics. It is also been used to improve the accuracy of text classifiers (Joulin et al., 2016).

2.2.2 *Multilingual Word Embeddings*

Recently, many efforts have been invested in multilingual embeddings; a complete *compendium* is provided by Ruder, Vulić, and Søgaard (2019). In general, acquiring word embeddings amounts to learning some mapping between bilingual resources, so to induce a shared space where words from both languages are represented in a uniform language-independent manner, "such that similar words (regardless of the actual language) have similar representations" (Vulić and Korhonen, 2016). A partially different and possibly complementary approach that may

[5] http://commoncrawl.org.

be undertaken is sense-oriented; it is best described as a graph-based approach, and proceeds by exploiting the information available in semantic networks such as WordNet and BabelNet. In the following we describe these two approaches in detail.

2.2.2.1 *Multilingual Embeddings Induction*

As regards as this first line of research, in most cases the alignment between two languages is obtained through parallel data, from which as close as possible vectorial descriptions are induced for similar words (see, e.g., the work by Luong, Pham, and Manning (2015)). A related approach consists in trying to obtain translations at the sentence level rather than at the word level, without employing word alignments (Chandar et al., 2014); the drawback is, of course, that large parallel *corpora* are required, which may be a too restrictive constraint on languages for which only scarce resources are available. In some cases (pseudo-bilingual training), Wikipedia has thus been used as a repository of text documents that are *circa* aligned (Vulić and Moens, 2015). Alternatively, dictionaries have been used to overcome the mentioned limitations, by translating the corpus into another language (Duong et al., 2016). Dictionaries have been used as seed lexicons of frequent terms to combine language models acquired separately over different languages (Faruqui and Dyer, 2014; Mikolov, Le, and Sutskever, 2013). Artetxe, Labaka, and Agirre (2018) propose a method using a dictionary to learn an embedding mapping, which in turn is used to iteratively induce a new dictionary in a self-learning framework by starting from surprisingly small seed dictionaries (a parallel vocabulary of aligned digits), that is used to iteratively align embedding spaces with performances comparable to those of systems based on much richer resources. A different approach consists in the joint training of multilingual models from parallel *corpora* (Coulmance et al., 2015; Gouws, Bengio, and Corrado, 2015). Also sequence-to-sequence encoder-decoder architectures have been devised, to train systems on parallel *corpora* with the specific aim of news translation (Hassan et al., 2018). Multilingual embeddings have been devised to learn joint fixed-size sentence representations, possibly scaling up to many languages and large corpora (Schwenk and Douze, 2017). Furthermore, pairwise joint embeddings (whose pairs usually involve the English language) have been explored, also for machine translation, based on dual-encoder architectures (Guo et al., 2018).

Conneau et al. (2018) propose a strategy to build bilingual dictionaries with no need for parallel data (MUSE), by aligning monolingual embedding spaces: this method uses monolingual corpora (for source and target language involved in the translation), and trains a discriminator to discriminate between target and aligned source embeddings; the mapping is trained through the adversarial learning framework, which is aimed at acquiring a mapping between the two sets such that translations are close in a shared semantic space. In the second step a synthetic dictionary is extracted from the resulting shared embedding space. The notion of shared semantic space is relevant to LessLex, which is however concerned with conceptual representations. Specifically, in our setting the sense inventory is available in advance, and senses (accessed through identifiers that can be retrieved by simply querying BabelNet) are part of a semantic network, and independent from any specific training *corpus*.

CONCEPTNET NUMBERBATCH. One multilingual resource that requires some additional focus is ConceptNet Numberbatch, which is heavily employed in the development of LessLex. ConceptNet Numberbatch (Robyn Speer, Chin, and Havasi, 2017) (CNN from now on) is a set of word embeddings that stems from the ConceptNet open data project. CNN is built by using an ensemble that combines ConceptNet, word2vec (Mikolov, Sutskever, et al., 2013), GloVe (Pennington, Socher, and Manning, 2014) by building on the retrofitting technique proposed by Faruqui, Dodge, et al. (2014). Retrofitting is a process that alters an existing set of word embeddings using a knowledge graph and, specifically, infers new vectors that are close to the original vectors but also close to their neighbors in the graph. In this application, the authors adopted the so called *expanded retrofitting* (Robert Speer and Chin, 2016). This can optimize the retrofitting objective function over a vocabulary that contains terms from the knowledge graph that do not appear in the embeddings vocabulary. This extension allows to fully exploit the multilingual connections in ConceptNet and also allows obtaining embeddings for non-English language terms that share the same semantic space as their English counterparts. In summary, after retrofitting to ConceptNet the pre-trained vector matrices of Glove and word2vec, the authors merged the two sets of embeddings by concatenating them and then reducing them to 300 dimensions via Singular Value Decomposition.

The final result is a set of multilingual embeddings indexed on words, meaning that words in different languages share a common semantic space which is informed by all of the languages. Such peculiarity, joint with the fact that CNN shows very good performances on most of the datasets for multilingual and cross-lingual word similarity, makes this resource a perfect candidate for further semantic extensions.

2.2.3 *Multi-Prototype, Sense-Oriented Embeddings*

Some works on word embeddings have dealt with the issue of providing different vectorial descriptions for as many senses associated to a given term. Such approaches stem from the fact that typical word embeddings mostly suffer from the so-called 'meaning conflation deficiency', which arises from representing all possible meanings of a word as a single vector of word embeddings. The deficiency consists of the "inability to discriminate among different meanings of a word" (Camacho-Collados and Pilehvar, 2018).

In order to account for lexical ambiguity, Reisinger and Mooney (2010) propose to represent terms as collections of prototype vectors; the contexts of a term are then partitioned to construct a prototype for the sense in each cluster. In particular, for each word different prototypes are induced, by clustering feature vectors acquired for each sense of the considered word. This approach is definitely relevant to LessLex for the attempt at building vectors to describe word senses rather than terms. However, one main difference is that the number of sense clusters K in our case is not a parameter (admittedly risking to inject noisy clusters as K grows), but it relies on the sense inventory of BabelNet, which is periodically updated and improved. The language model proposed by E. H. Huang et al. (2012) exploits both local and global context, that are acquired through a joint training objective. In particular, word representations are computed while learning to discriminate the next word, given a local context composed of a short sequence of words, and a global context composed by the whole document where the word sequence occurs. Then, the collected context representations are clustered, and each occurrence of the word is labelled with its cluster, and used to train the representation for that cluster. The different meaning groups are thus used to learn multi-prototype vectors, in the same spirit as in the work by Reisinger and Mooney (2010). Also relevant, the work by Neelakantan et al. (2014) proposes an extension to the Skip-gram model to efficiently learn multiple embeddings per word type: interestingly enough, this approach obtained state-of-the-art results in the word similarity task. The work carried out by T. Chen et al. (2015) directly builds on a variant of the Multi-Sense Skip-Gram (MSSG) model by Neelakantan et al. (2014) for context clustering purposes. Namely, the authors propose an approach for learning word embeddings that relies on WordNet glosses composition and context clustering; this model achieved state-of-the-art results in the word similarity task, improving on previous results obtained by E. H. Huang et al. (2012) and by X. Chen, Z. Liu, and Sun (2014).

Another resource that is worth mentioning is SenseEmbed (Iacobacci, Pilehvar, and Navigli, 2015); the authors propose here an approach for obtaining continuous representations of individual senses. In order to build sense representations, the authors, exploited Babelfy (Moro, Raganato, and Navigli, 2014) as word sense disambiguation system on the September-2014 dump of the English Wikipedia.[6] Subsequently the word2vec toolkit has been employed to build vectors for 2.5 millions of unique word senses.

[6] http://dumps.wikimedia.org/enwiki/.

NASARI. Another project we need to describe in some detail is NASARI (Camacho-Collados, Pilehvar, and Navigli, 2016). NASARI builds on BabelNet and Wikipedia; it is indexed on Babel synset IDs and specifically describes nouns and named entities, but not other grammatical categories. NASARI has been released in three different versions: *lexical*, *embedded* and *unified*. The basic notion on which the resource is built is the one of *lexical specificity*, which is a statistical measure that computes the most significant words for a given text based on the hypergeometric distribution. More precisely, given a *reference corpus* $\mathcal{RC}$ and a *sub-corpus* $\mathcal{SC}$ (with $\mathcal{SC} \subset \mathcal{RC}$) the lexical specificity computes the weights for each word by contrasting the frequencies of that word across $\mathcal{SC}$ and $\mathcal{RC}$.

In order to build the *lexical* NASARI vector for a Babel synset s, the authors consider the whole Wikipedia as reference corpus $\mathcal{RC}$, while the sub-corpus $\mathcal{SC}$ is built by considering *i*) the page representing s in Wikipedia; *ii*) all the pages with outgoing links to the page of s in Wikipedia; and *iii*) all the pages representing the hypernyms and hyponyms of s in Wikipedia. Lexical specificity is then computed to obtain the weights of the words in $\mathcal{SC}$ that will finally constitute the vector of s. For instance the final lexical vector of the concept *Admiration* (`bn:00001454n`) looks as follows:

```
bn:00001454n Admiration awe_173.29 admiration_83.41
emotion_81.1 descartes_75.99 elevation_68.7
passion_64.63 wonder_40.44 ...
```

These lexical vectors have then been leveraged to build the *embedded* version of NASARI (NASARIE for brevity). Given any set of pre-existing word embeddings and a NASARI lexical vector for a sense s, it is possible to build its corresponding NASARIE vector by simply averaging the word embeddings associated to the terms appearing in its lexical vector. To fully take advantage of the NASARI lexical structure, the average computed is actually weighted by taking into consideration the rank (i.e., the weight) of the word as it appears in the lexical vector s. The released NASARIE is built by using the word2vec vectors trained over the Google News dataset: the final result is a set of $2.9M$ vectors that also share the same semantic space of word2vec, so that their representations can be used to compute semantic distances between any two such vectors.

The last version of NASARI is called NASARI *unified* and it is basically a *disambiguated* version of NASARI lexical: in this version each vector contains Babel synsets instead of words. Starting from the sub-corpus $\mathcal{SC}$, the lexical specificity is computed among clusters of hyponyms (i.e., senses sharing the same hypernym) instead of single words: this clustering of sibling words into a single cluster represented by their common hypernym transforms a lexical space into a unified semantic which has multilingual synsets as dimensions. As example we can look at the resulting unified vector for the concept *Admiration* (`bn:00001454n`):

```
bn:00001454n Admiration bn:00033963n_91.2 bn:00030581n_71.59
bn:00001455n_69.27 bn:00016845n_26.16
bn:00061984n_21.12 ...
```

In this work we will make use of the NASARI *unified* (simply referred to as NASARI) and *embedded* (referred to as NASARIE).

2.3 WORD AND SENSE SIMILARITY

In this section we introduce the two main tasks that were employed to evaluate our resources COVER and LessLex. Further details on the considered datasets and the adopted approaches for the resolution of the tasks will be provided in the respective evaluation sections of Chapter 3 and Chapter 4.

2.3.1 *Concept Similarity*

The concept similarity task has a rather simple definition: given a pair of *concepts* in input, the system has to estimate a similarity score (in some defined range) between the two. Concepts are usually provided as unique sense identifiers in a given sense inventory such as the one of WordNet or BabelNet. The conceptual similarity task is a long-standing task adopted for the evaluation of lexical resources (Miller and Charles, 1991; Resnik, 1995; Richardson, Smeaton, and Murphy, 1994; Wu and Palmer, 1994). Depending on the type of lexical resource at hand, the access to its knowledge can vary. For vectorial resources that are indexed on *senses* (e.g., NASARI, COVER, LessLex) the task is usually straightforwardly solved by computing the cosine similarity between the two vectors matching the input concepts.[7] On the contrary, for lexical resources such as word embeddings (e.g., GloVe, CNN) some additional steps are required in order to link the input concepts to the internal terminological representation of the resource at hand. Other types of knowledge representation such as taxonomies can however rely

[7] The senses inventories between the resource and the dataset must match or be linked.

on their inner structure to compute the similarity (Leacock, Miller, and Chodorow, 1998; Hansen A Schwartz and Gomez, 2008; Wu and Palmer, 1994), possibly also relying on information content (Jiang and Conrath, 1997; Resnik, 1998).

2.3.2 *Word Similarity*

The word similarity is a more general variant of the concept similarity task in which two *words* are provided as input instead of concepts. In this settings, word-indexed resources (e.g., word embeddings) are facilitated in the resolution of the task, while conceptual resources need to perform some kind of *disambiguation* of the words to access their inner representations for the input. The rationale is that each term works as the context for the other one (e.g., in the pairs ⟨'fork','system call'⟩, and ⟨'fork','river'⟩). In particular, to compute the semantic similarity between a term pair, a variant of a general disambiguation approach formerly proposed in Pedersen, Banerjee, and Patwardhan (2005) can be adopted. Such disambiguation approach is defined as follows:

GIVEN: a pair $\langle w_t, C\rangle$, where w_t is the term being disambiguated, and C is the context where w_t occurs, $C = \{w_1, w_2, \ldots, w_n\}$, with $1 \leq t \leq n$; also, each term w_i has m_i possible senses, $s^i_1, s^i_2, \ldots, s^i_{m_i}$.

FIND: one of the senses from the set $\{s^t_1, s^t_2, \ldots, s^t_{m_t}\}$ as the most appropriate sense for the target word w_t.

The basic idea is to compute the semantic similarity as a function maximizing the similarity between each two senses (corresponding to the target term and to all terms in the context C) by finding the best sense s_h^t disambiguating w_t where h is computed as:

$$h = \operatorname*{argmax}_{m_i=1}^{m_t} \left[\sum_{w_j \in C, j \neq t} \max_{k=1}^{m_j} \otimes \left(s_i^t, s_k^j \right) \right] \tag{2.1}$$

where $\otimes$ is implemented by some similarity metrics, such as the cosine similarity. When dealing with the word similarity task, the context of each word is shrunk to the other word in the pair, so, the most adopted approach (Budanitsky and Hirst, 2006; Pilehvar and Navigli, 2015) (often refereed to as *max-similarity*), can be formulated as follows: given two terms w_1 and w_2, each with an associated list of senses $s(w_1)$ and $s(w_2)$, their semantic similarity can be computed as:

$$\text{sim}(w_1, w_2) = \max_{c_1 \in s(w_1), c_2 \in s(w_2)} [\text{sim}(c_1, c_2)] \tag{2.2}$$

where $\text{sim}(c_1, c_2)$ refers to the similarity computed among conceptual representations inside the semantic knowledge base.

SIMILARITY VS RELATEDNESS A clarification must be made between the two notions of *semantic similarity* and *semantic relatedness*. The former is a subset of the latter, but in some contexts the two terms can be used interchangeably. Two senses can be assessed based on their similarity axis if there is a synonymy (e.g., *bank–trust company*), hyponymy/hypernymy, antonymy, or troponymy relation between them (e.g., *veichle–car*), while the semantic relatedness is based on shallower lexical relationships such as, for instance, meronymy (e.g., *handle–door*) (Budanitsky and Hirst, 2006; Mohammad and Hirst, 2012). Interestingly, some of the benchmark datasets on the similarity task disregard this key difference, an issue that has been pointed out in e.g. Agirre et al. (2009). In this work we will also provide a detailed analysis of the SemEval-2017 dataset concerning this aspects in Section 3.2.3.

Part II

RESOURCES

3

COVER

In this chapter the COVER ('COmmon-sense VEctorial Representation') resource will be introduced, along with COVERAGE ('COVER Automatic GEnerator'), the algorithm devised to built it (Lieto, Mensa, and Radicioni, 2016a; Mensa, Radicioni, and Lieto, 2017b, 2018).

3.1 BUILDING COVER

3.1.1 *Enriching Common-Sense Knowledge*

In the recent years a lot of effort has been put into the development of resources aimed at providing systems with human-level competence in understanding text documents. However, one main type of information that has been rarely taken in consideration is *common-sense knowledge*. Common-sense can be defined as a widely accessible and elementary form of knowledge (Minsky, 2000), that can also be seen as prototypical knowledge (Rosch, 1975). For instance, in considering the concept of *dish* its common-sense traits could be that it usually made of ceramic and it is used to contain and serve foods.[1] This kind of information can become

[1] "When people communicate with each other, they rely on shared background knowledge to understand each other: knowledge about the way objects relate to each other in the world, people's goals in their daily lives, the emotional content of events or situations. This 'taken for granted' information is what we call common sense – obvious things people normally know and usually leave unstated" (Cambria, Robyn Speer, et al., 2010, p.15).

very relevant in settings where artificial agents need to complement more structured information (such as, e.g., about the chemical composition or taxonomic information) with common-sense aspects.

The richest resource providing common-sense knowledge is ConceptNet (Section 2.1) whose knowledge is, however, concerned with terms rather then senses. In particular, due to the heterogeneous representation provided by its nodes it can be very difficult to extract all the relevant information pertaining to specific concept.

Under this premise, the COVER resource has been developed in order to enrich the common-sense provided by ConceptNet and make it more precise by linking it to the BabelNet sense inventory. The final result is a collection of vectors indexed by sense that provide common-sense knowledge. Such vectors have been employed to tackle a vast variety of tasks.

3.1.2 *Concepts Representation in* COVER

The structure of the vectors in COVER is based on 45 relationships available in ConceptNet[2]. In order to build this subset we started from the complete list of ConceptNet relationships and we pruned those that were either not suitable for our task (e.g., ExternalURL, dbpedia/language) or extremely specific (e.g., dbpedia/occupation, dbpedia/genus). The final set includes relationships that describe

[2] InstanceOf, RelatedTo, IsA, AtLocation, dbpedia/genre, Synonym, DerivedFrom, Causes, UsedFor, MotivatedByGoal, HasSubevent, Antonym, CapableOf, Desires, CausesDesire, PartOf, HasProperty, HasPrerequisite, MadeOf, CompoundDerivedFrom, HasFirstSubevent, dbpedia/field, dbpedia/knownFor, dbpedia/influencedBy, dbpedia/influenced, DefinedAs, HasA, MemberOf, ReceivesAction, SimilarTo, dbpedia/influenced, SymbolOf, HasContext, NotDesires, ObstructedBy, HasLastSubevent, NotUsedFor, NotCapableOf, DesireOf, NotHasProperty, CreatedBy, Attribute, Entails, LocationOfAction, LocatedNear.

```
Exemplar bn:00069619n (school, university, academy)

HasA    [period, classroom]
PartOf  [school system, academia]
IsA     [establishment, building, educational institution, ...]
UsedFor [pedagogy, degree, learning, education, ...]
...
```

Figure 3.1: A portion of the COVER vector for the *school* concept. For sake of the readability the sense identifiers filling the dimensions have been replaced with their corresponding lexicalization.

both *relatedness* and *similarity* traits, allowing COVER to be a complete resource that can be tailored to specifically compute semantic relatedness or similarity. In our evaluation of the resource we did not however focus on this aspect, but rather tried to establish the quality of the resource in its entirety.

Each ConceptNet relationship has been mapped to a correspondent vector dimension in COVER. Each dimension is filled by a set of values that are concepts themselves, each identified through its BabelSynset ID, taken from BabelNet. So a concept c_i has a vector representation $\vec{c_i}$ that is formally defined as

$$\vec{c_i} = [s_1^i, .., s_N^i], \tag{3.1}$$

where each s_h^i is the set of concepts filling the dimension $d_h \in D$. Each s can contain an arbitrary number of values, or be empty. As an example, Figure 3.1 shows the vector representing the concept `bn:00069619n` (*school)*.

3.1.3 *Feeding the System: the* CLOSEST *Algorithm*

One of the key aspects of a resource such as COVER is the amount of knowledge (concepts) that the resource is able to represent. Since COVERAGE–the algorithm that generates COVER– takes in input a sense represented as Babel synset ID and produces a COVER vector for it, we had to collect a set of terms and associated senses to fed to the system. Ideally such set could be equivalent to the whole sense inventory represented in BabelNet, however, it is acknowledged that too fine-grained semantic distinctions may be unnecessary and even detrimental in many tasks (Palmer, Babko-Malaya, and Dang, 2004): for this reason we developed the CLOSEST algorithm (Lieto, Mensa, and Radicioni, 2016b), which is employed to obtain the set of senses that will constitute the input of COVERAGE. Specifically, CLOSEST accesses BabelNet and produces more coarse-grained sense inventories, based on a simple heuristics that builds on the notions of *availability* and *salience* of words and phrases (Vossen and Fellbaum, 2009). In summary, thanks to CLOSEST we can detect, given a set of terms, their most relevant senses and then feed them to COVERAGE.

We start from all of the English *nouns* from the Corpus of Contemporary American English (COCA) (Davies, 2009), which is a corpus covering different genres, such as spoken, fiction, magazines, newspaper and academic.[3] This set of terms is the input to CLOSEST: the selection of the most relevant senses is performed upon the hypothesis that more central senses are more richly represented in encyclopedic resources. We

[3] http://corpus.byu.edu/full-text/.

estimate the centrality of a concept by looking at its generality and connectivity w.r.t. the others.

The CLOSEST algorithm works as follows: given an input term t, the set of senses $S = \{s_1, s_2, \ldots, s_n\}$ that could be possibly associated to t is retrieved: such set is obtained by directly querying BabelNet. The final set of relevant senses is then obtained by incrementally filtering S, via two main strategies:

1. *LS-Pruning.* Pruning of less salient senses: senses with associated poor information are eliminated. The salience of a given sense is determined by inspecting its NASARI unified vector (Section 2.2.3);

2. *OL-Pruning.* Pruning of overlapping senses: for each two senses with significant overlap (a function of the number of features shared in the corresponding NASARI unified vectors), the less salient sense is pruned.

In summary, CLOSEST inspects the NASARI unified vectors of each candidate sense of a term to determine if it is relevant or not. The following sections illustrate in detail how the LS-Pruning and OL-Pruning phases take place.

3.1.3.1 *LS-Pruning*

The analysis of the set S of candidate senses of t starts by collecting each NASARI vector $\vec{v}_{ts}$ that represents the specific sense s for the term t. The first pruning strategy is merely based on the size of this vector: since shorter vectors often refer to peripheral senses, if the vector size is less then a fixed α quantity, it is pruned. The subsequent pruning strategy requires the computation of:

- $\overline{W}(\vec{v}_{ts})$ which indicates the weight of the candidate sense vector $\vec{v}_{ts}$, and is computed as the average of all the weights associated to its features (i.e., synsets).
- $L(\vec{v}_t)$ which represents the longest vector among all the candidate senses for t;
- $H(\vec{v}_t)$ which represents the heaviest vector in S among all the candidate senses for t.

The definitions for these measures are illustrated in Equations 3.2–3.4.

$$L(\vec{v}_t) = \arg\max_{s \in S} \left(\mathrm{len}(\vec{v}_{ts})\right) \tag{3.2}$$

$$\overline{W}(\vec{v}_{ts}) = \frac{1}{\mathrm{len}(\vec{v}_{ts})} \cdot \sum_j w_{sj} \tag{3.3}$$

$$H(\vec{v}_t) = \arg\max_{s \in S} \left(\overline{W}(\vec{v}_{ts})\right). \tag{3.4}$$

The decision on whether to prune or not a vector is based on a simple criterion: $\vec{v}_{ts} \in S$ is pruned if both its length is below a given fraction of the length of the longest one $L(\vec{v}_t)$, *and* its weight is lower than a given fraction of the heaviest one, $H(\vec{v}_t)$. The parameter settings adopted by our pruning rules are illustrated in Table 3.1.

Table 3.1: The senses pruning conditions in CLOSEST.

	condition	values	pruning phase
prune $\vec{v}_{ts}$ IF	$\mathrm{len}(\vec{v}_{ts}) \le \alpha$	$\alpha = 5$	*LS-Pruning*
	$\left(\frac{\mathrm{len}(\vec{v}_{ts})}{L(\vec{v}_t)} < \beta\right)$ AND $\left(\frac{\overline{W}(\vec{v}_{ts})}{\overline{W}(H(\vec{v}_t))} < \gamma\right)$	$\beta, \gamma = .40$	*LS-Pruning*
	$Ovl(\vec{v}_{ts}, \vec{v}_{tu}) \ge \delta$	$\delta = .20$	*OL-Pruning*

3.1.3.2 *OL-Pruning*

The second phase of the algorithm aims at the detection of overlapping senses. We rely once again on NASARI to further prune S. The overlap between two vectors $Ovl(\vec{v}_{ti}, \vec{v}_{tj})$ is computed as a fraction of the length of the shortest vector between the two considered, as indicated in Equation 3.5.

$$Ovl(\vec{v}_{ti}, \vec{v}_{tj}) = \frac{\vec{v}_{ti} \cap \vec{v}_{tj}}{\mathrm{len}(shortest(\vec{v}_{ti}, \vec{v}_{tj}))} \tag{3.5}$$

The overlapping is checked for every pair $\langle \vec{v}_i, \vec{v}_j \rangle$ (with $i \neq j$), and when an overlap is detected higher than a fixed threshold (see Table 3.1), the shortest vector between the two is pruned. The rationale is that this possibly less relevant sense is already included in (and therefore represented by) a more relevant sense.

The tuning of the parameters has been performed by manually examining a substantial amount of cases, however, depending on the application for which CLOSEST is used they could be set differently. Very specific senses could be retained by increasing β and γ, while the ability to preserve different meaning nuances can be obtained by augmenting δ. For instance the term *time* presents 48 senses mostly representing magazines, movies, songs, and music albums: all of these senses are presently pruned by CLOSEST but they could be preserved for instance in the development of a NER algorithm.

Once the whole CLOSEST algorithm is applied to each COCA term we have a set of senses that can be finally used as input to the COVERAGE algorithm.

3.1.4 *The* COVERAGE *Algorithm*

Once the CLOSEST algorithm has determined which senses must be present in the COVER resource, the corresponding vectors are generated via the COVER algorithm. The purpose of the COVERAGE algorithm is to build a COVER vector given as input a certain concept c, provided as Babel synset ID. The algorithm consists of two main steps:

1. **Semantic Extraction**:
   - *Extraction*: all nodes representing any lexicalization of c in ConceptNet are retrieved and all the relevant terms connected to such nodes are triggered and placed in the set of extracted relevant terms T (more about relevance criteria later on).
   - *Concept Identification*: all terms $t \in T$ are disambiguated into their corresponding Babel synset ID; this step amounts to translating T into the set of relevant extracted concepts C.
2. **Vector Injection**: each concept $c_i \in C$ is injected into its vector representation $\vec{c}$ by exploiting the relationship formerly connecting c_i to c in ConceptNet.

Figure 3.2 illustrates the general outline of COVERAGE. In the following section we will explore the algorithm in depth by following its execution upon the concept $c =$ `bn:00035902n`, that is *Fork* intended as “the utensil used for eating or serving food”.

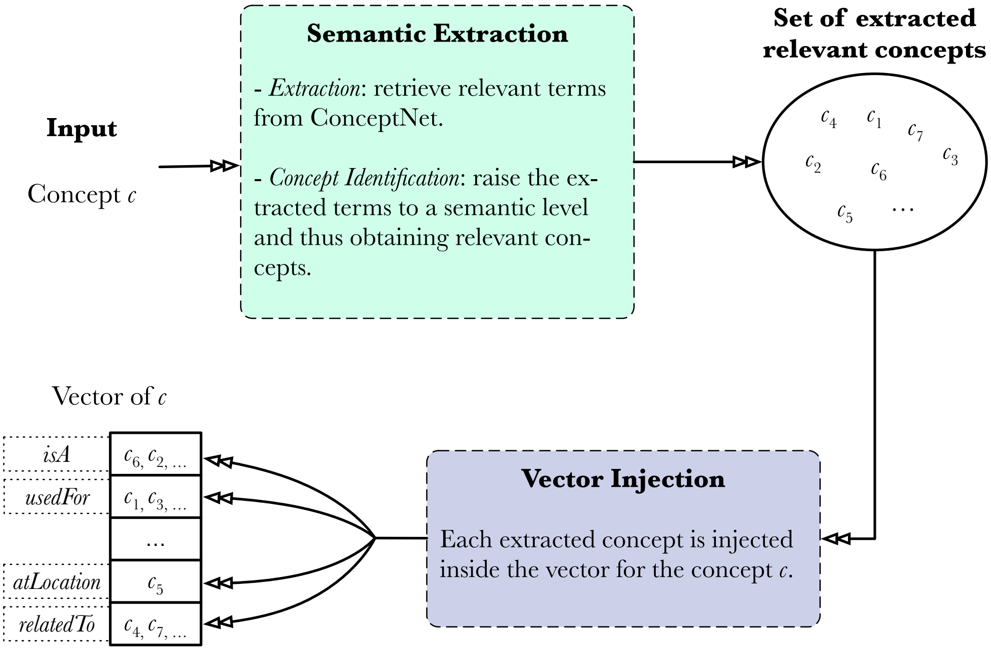

Input
Concept c
Semantic Extraction
- *Extraction*: retrieve relevant terms from ConceptNet.
- *Concept Identification*: raise the extracted terms to a semantic level and thus obtaining relevant concepts.
Set of extracted relevant concepts
c_4 c_1 c_7 c_3 c_2 c_6 c_5 ...
Vector of c
isA
$c_6, c_2, ...$
usedFor
$c_1, c_3, ...$
...
atLocation
c_5
relatedTo
$c_4, c_7, ...$
Vector Injection
Each extracted concept is injected inside the vector for the concept c.

Figure 3.2: The outline of the COVERAGE algorithm.

3.1.4.1 *Semantic Extraction*

Purpose of this first portion of the algorithm is to collect the set C of *relevant concepts* associated to the input concept c, which will constitute the content of the final vector $\vec{c}$.

We start by straightforwardly retrieving the NASARI (unified) vector associated to c, which is naturally indexed as `bn:00035902n`. This vector will serve as the semantic root upon which the entire process revolves. We then access all ConceptNet nodes that could potentially represent the concept, and specifically we retrieve those that represent one of c lexicalizations (which are obtainable via BabelNet). In our *Fork* case, we look for the nodes *Fork*, *King of utensils*, *Pickle fork*, *Fish fork*, *Dinner fork*, *Chip fork* and *Beef fork* in ConceptNet. All of the connections starting from these nodes are then put together and examined: as explained in Section 2.1, the lack of a semantic level in ConceptNet requires to filter out the inappropriate connections. In other words, since

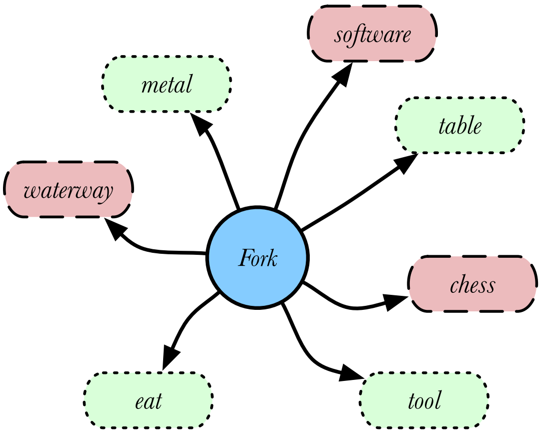


Figure 3.3: Each term connected to the ConceptNet node *Fork* is inspected to determine whether it is relevant (dotted contour) or not (dashed contour) for the sense conveyed by the input concept c. While the dotted nodes are relevant because they are referring to *Fork* as the "kitchen utensil" —that is, the sense of c—, the dashed ones refer to *Fork* as the system call for creating processes (*software* node), as the chess move (*chess* node), or as the bifurcation of a watercourse (*waterway* node).

we have retrieved nodes based on c lexicalizations, we have extracted connections (or equivalently terms) that refer to those lexicalizations in any of their meanings and not only the meaning conveyed by c. To determine if an extracted term t is *relevant* for c or not, we developed the following criteria:

Definition 3.1.1 (Relevance Criteria)**.** An extracted term t is considered relevant for the concept c if either: *i)* t is included in at least one of the synsets listed in the NASARI vector representation for c; or *ii)* at least β nodes directly connected to t in ConceptNet can be found in the synsets that are part of the NASARI vector representation for c.

Figure 3.3 illustrates the *Fork* node in ConceptNet and its relevant/non relevant connected nodes. The rationale underlying the relevance criteria is explained by the fact that since the NASARI unified vector of c contains concepts (along with their lexicalizations) semantically close to c, the presence of t (first condition) or β terms from its ConceptNet neighborhood (second condition) in such vector guarantees that t is somehow related to c, and it can be thus considered as relevant.

Once the relevance detection is performed, all the relevant terms extracted from all the ConceptNet nodes that we previously collected are put together in the set T. In the *Fork* example, the resulting set is:

$$T = \{\textit{plate, tool, food, utensil, silverware, table, metal knife, spoon, eat}\} \tag{3.6}$$

Once the set T of relevant terms is built, each of the terms has to be disambiguated by assigning a Babel synset ID to it. Such process is performed during the *Concept Identification* step. The behavior of the Concept Identification step depends on how a term $t \in T$ has been detected as relevant. More precisely, if t was detected relevant via the first condition, it must appear inside the NASARI unified vector of c, and so we can directly retrieve its Babel synset ID. On the other hand, if t was recognized as relevant via the second condition its Babel synset ID cannot be directly obtained. In such case, all of the possible candidate senses of t are collected via BabelNet; for each candidate we then access its NASARIE vector and we compute its cosine similarity w.r.t. the NASARIE vector of c: the sense with the smaller distance is selected. A threshold system is also put in place, so, the similarity of the close candidate must surpass a fixed quantity. Figure 3.4 illustrates this process for the *Fork* example.

Once the Concept Identification is completed, the term t is enriched with its Babel synset ID and included in the set of the relevant extracted concepts C.

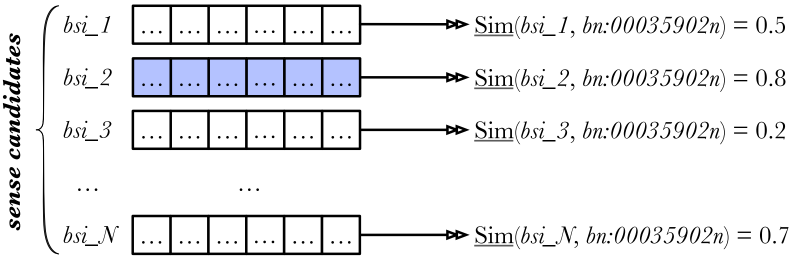

sense candidates
bsi_1
bsi_2
bsi_3
…
bsi_N
… … … … … …
Sim(bsi_1, bn:00035902n) = 0.5
Sim(bsi_2, bn:00035902n) = 0.8
Sim(bsi_3, bn:00035902n) = 0.2
Sim(bsi_N, bn:00035902n) = 0.7

Figure 3.4: The similarity between NASARIE candidate vectors and the vector of *Fork* (`bn:00035902n`) is computed. The highlighted vector is selected since its similarity with the *Fork* vector obtained the highest score (and it surpasses the required threshold).

For the experimentation illustrated in Section 3.2, the β parameter and the similarity threshold are set to 2 and 0.6 respectively. The tuning of these parameters has been performed by examining both the ConceptNet neighborhood and the disambiguated senses returned by the system on a set of randomly chosen samples. We prioritized the correctness over the completeness of the extracted concepts, so we set rather restrictive parameters.

3.1.4.2 *Vector Injection*

The second and last phase of the COVERAGE system consists in injecting the values of C inside the empty structure of a COVER vector, thus obtaining $\vec{c}$, the populated vector for c. Since each concept $c_i \in C$ has been extracted from ConceptNet, we still have access to the relationship that was connecting it to one of the lexicalizations of c (extraction step). Thanks to the fact that the COVER vectors dimensions are basically a selection of ConceptNet relationships (Section 3.1.2), the Vector Injection amounts to coherently place inside $\vec{c}$ each $c_i \in C$ into the dimension corresponding to the relationship that was linking c_i to c in ConceptNet. Figure 3.4 illustrates the Vector Injection for the *Fork* example.

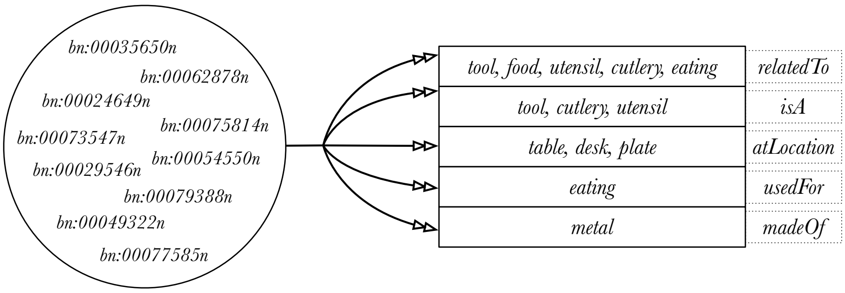


Figure 3.5: All the concepts in C are injected into the vector for *Fork*. The concepts identifiers in the vector have been replaced with their lexicalization in order to make the image human readable.

The next section will present some general figures regarding the data fed to COVERAGE and the resulting set of vectors which is COVER.

3.1.5 COVER *Statistics*

We now present some figures and statistics regarding the computation of COVERAGE, including the size of the lexical base taken as input, some numbers on retrieved (and discarded) concepts, and a final quantitative description of the amount of information finally encoded in COVER.

INPUT. The concepts fed to COVER were obtained by executing the CLOSEST algorithm upon the nouns of Corpus of Contemporary American English. In detail, CLOSEST took $27,006$ terms in input, and returned $40,816$ concepts in output, meaning that we obtained on average 1.5 senses for each term. These concepts were then fed to the COVERAGE system. It is important to note that this selection of senses does not affect the content of the single vectors but rather the final amount of representations contained in the resource. For this reason we selected COCA, which is a large and general corpus of terms.

Before executing the system, we furtherly pruned the set of concepts by removing duplicated concepts ($8,867$) or concepts for which we could not find a NASARI vector (112). The remaining $31,837$ concepts constitute the input of COVERAGE. The size of the resources employed all throughout this process is reported in Table 3.2.

Table 3.2: Information contained in NASARI and ConceptNet, and used as the starting point to build COVER.

Resource	Size
NASARI/NASARIE *vectors*	2,868,176
ConceptNet *assertions*	4,227,874
ConceptNet *nodes*	859,932

SEMANTIC EXTRACTION. During the *Semantic Extraction* phase, a total of $4,324,971$ terms were extracted from ConceptNet (on average, 135.85 per input concept), but only 42.9% of them (overall $1,856,888$) were found relevant, resulting in an average cardinality of T for each input of 58.32. The disambiguation performed during the Concept Identification was successful for the 32.61% of the relevant terms, thereby resulting in a total of $605,450$ extracted relevant concepts (the average cardinality of the bag of concepts C was then 19.02). We note that roughly two thirds of the concept identification failures were due to the violation of the concept similarity threshold. This threshold is indeed a very sensitive parameter that allows for the tuning of the amount of noise (vs. completeness) featuring the resource: e.g., by setting the similarity threshold to 0.5 instead of 0.6, the average cardinality of C raises to 25.86 (which directly compares with the actual value, 19.02).

VECTOR INJECTION. All of the concepts in C were used during the *Vector Injection* phase, since COVERAGE only loads the ConceptNet relationships that are already included in the COVER schema. Therefore, the resulting average number of values per concept corresponds to the average cardinality of C (19.02). This figure was then increased by adding the first 5 elements contained in the NASARI vector for the input concept in its RELATEDTO dimension, bringing the average population of the vectors to 23.97. More precisely, half vectors contain 5 to 20 values, while only 0.5% vectors are filled by less than five values.

DIMENSIONS. The most populated dimensions are RELATEDTO, SYNONYM, ISA, HASCONTEXT, ANTONYM, FORMOF and DERIVEDFROM: this distribution closely approaches the distribution of information contained in ConceptNet (Table 3.3).

FAILURE CASES. The COVERAGE system obtained an empty set C for $4,786$ concepts out of the $31,837$ provided as input. In such cases, the resulting vectors for such concepts contain exclusively values that were automatically taken from NASARI and injected into the RELATEDTO dimension. More in detail, in most failure cases (namely, $4,570$) the system either could not detect any extracted relevant term, or it could not disambiguate any of the extracted terms. For instance, the input *recantation* produced only *recall* as extracted term. However, the similarity between these two concepts was under the threshold β, therefore, *recall* couldn't be accepted and the C set for *recantation* resulted empty. In the remaining 216 cases, it was not possible to find

Table 3.3: The 15 most populated dimensions in ConceptNet 5.5.0.

Relationship	Number of associations	% of associations
RelatedTo	1,449,431	51.25%
FormOf	273,560	09.67%
IsA	247,387	08.75%
Synonym	237,772	08.41%
HasContext	177,677	06.28%
DerivedFrom	116,243	04.11%
UsedFor	42,443	01.50%
SimilarTo	29,480	01.04%
AtLocation	28,960	01.02%
CapableOf	26,354	00.93%
HasSubevent	25,896	00.92%
HasPrerequisite	23,493	00.83%
EtymologicallyRelatedTo	20,723	00.73%
Antonym	19,967	00.71%
Causes	17,088	00.60%

a ConceptNet node for the input concept. We observed that the vast majority of this concepts contained a dash (e.g., *tete-a-tete*, *god-man*, *choo-choo*). A further improvement would consist in the removal of such dashes in order to detect a suitable ConceptNet node for this kind of inputs.

The download link for COVER can be found in Appendix B.

3.2 EVALUATING COVER

The main evaluation of COVER has been carried out on the word and concept similarity task, introduced in Section 2.3. To these ends we designed the MeRaLi system, which computes semantic similarity at both sense and word level by specifically relying on COVER. MeRaLi was originally presented in the frame of the Sem-Eval 2017 campaign on Multilingual and Cross-lingual Semantic Word Similarity (Mensa, Radicioni, and Lieto, 2017a); the experimentation was then extended by employing an updated version of COVER.

In this section we first illustrate the similarity metrics implemented by the MeRaLi system; we then introduce the data sets used for testing, and provide the results along with their discussion.

3.2.1 *The Similarity Measure*

As previously mentioned, the concept similarity task consists in the estimation of a similarity score between two given concepts.

When using COVER, the task can be actually cast to a vector-comparison problem under the rationale that the two vectors representing the input concepts (as depicted in Equation 3.1) are similar proportionally to the amount of information that they share. For instance, two objects that share the same material (MadeOf), use (UsedFor) and location (LocatedAt) are probably very similar.

In detail, given two input concepts c_i and c_j, after the retrieval of the corresponding COVER vectors $\vec{c}_i$ and $\vec{c}_j$, we compute their similarity by computing, dimension by dimension, the set of shared values between $\vec{c}_i$ and $\vec{c}_j$. Then, the similarity score obtained over each dimension is combined by obtaining an overall similarity score, that is our final output. So, given N dimensions in each vector, the similarity value, $\mathrm{sim}(\vec{c}_i, \vec{c}_j)$, should be ideally computed as:

$$\mathrm{sim}(\vec{c}_i, \vec{c}_j) = \frac{1}{N} \sum_{k=1}^{N} |s_k^i \cap s_k^j|, \tag{3.7}$$

where s_k^i represents the set of values assigned to the dimension k for the i vector. However, this formulation resulted to be too naïve. In fact, the information available in COVER is not evenly distributed, that is, it may happen that a given dimension is filled with many values (concepts) in the description of a given concept, but the same dimension may be empty in the description of another one. It was hence necessary to refine the above formula to tune the balance between the amount of information available for the concepts at stake: *i)* at the individual dimension level, to balance the number of concepts that characterize the different dimensions; and *ii)* across dimensions, to prevent the computation from being biased by more richly defined concepts (i.e., those with more dimensions filled). Both *desiderata* are satisfied by the Symmetrical Tversky's Ratio Model (Jimenez et al., 2013) (which is a symmetrical reformulation for the Tversky's ratio model (Tversky, 1977)),

$$\mathrm{sim}(\vec{c}_i, \vec{c}_j) = \frac{1}{N^*} \cdot \sum_{k=1}^{N^*} \frac{|s_k^i \cap s_k^j|}{\beta\,(\alpha a + (1-\alpha)\ b) + |s_k^i \cap s_k^j|} \tag{3.8}$$

where $|s_k^i \cap s_k^j|$ counts the number of shared concepts that are used as fillers for the dimension d_k in the concept $\vec{c}_i$ and $\vec{c}_j$, respectively; a and b are defined as

$$a = \min(|s_k^i - s_k^j|, |s_k^j - s_k^i|),$$
$$b = \max(|s_k^i - s_k^j|, |s_k^j - s_k^i|);$$

while N^* counts the dimensions actually filled with at least two concepts in both vectors. This formula allows tuning the balance between cardinality differences (through the parameter α), and between $|s_k^i \cap s_k^j|$ and $|s_k^i - s_k^j|, |s_k^j - s_k^i|$ (through the parameter β).[4]

Finally, when dealing with the word similarity task rather then the conceptual similarity, the disambiguation can be performed by exploiting the max-similarity approach (Equation 2.2), implemented by using the Symmetrical Tversky's Ratio Model as the sim function.

3.2.2 *Experimental Setting and Procedure*

DATA SETS. The performance of the MeRaLi system has been assessed over four standard data sets. We considered three data sets for conceptual similarity at the *sense* level,[5] namely the RG (Rubenstein and Goodenough, 1965), MC (Miller and Charles, 1991) and WS-Sim data set, which was first designed for conceptual relatedness in (Finkelstein et al., 2001) and then partially annotated with similarity judgments (Agirre et al., 2009). Additionally, we considered a fourth dataset released in the frame of the SemEval-2017 campaign on Multilingual and Cross-lingual Semantic Word Similarity, and concerned with the computation of the conceptual similarity at the *word* level (Camacho-Collados, Pilehvar, Collier, et al., 2017a).

More in detail, the MC data set actually contains 28 pairs, that are a subset of the RG data set, containing 65 sense pairs. The WS-Sim data set is composed of 97 sense pairs, and the Sem-Eval 2017 data set consists of 500 word pairs. The last data set is the most challenging, since

[4] The parameters α and β were set to 0.8 and 0.2 for the experimentation. The tuning has been performed by looking at the result obtained with different combinations of α and β on the SemEval dataset.

[5] Publicly available at the URL http://www.seas.upenn.edu/~hansens/conceptSim/.

it hosts word pairs involving entities. It is challenging also for human common sense in many ways, since it includes pairs such as ⟨Si-o-seh pol, Mathematical Bridge⟩ and ⟨Mount Everest, Chomolungma⟩.

EVALUATION METRICS. The quality of the scores provided in output by the MeRaLi system have been assessed through Pearson's r and Spearman's ρ correlations, that are usually adopted for the conceptual similarity task. The Pearson r value captures the linear correlation of two variables as their covariance divided by the product of their standard deviations, thus basically allowing to grasp differences in their values, whilst the Spearman ρ correlation is computed as the Pearson correlation between the *rank* values of the considered variables, so it is reputed to be best suited to assess results in a similarity ranking setting where relative scores are relevant (Pilehvar and Navigli, 2015; Hansen Andrew Schwartz and Gomez, 2011). Furthermore, we recorded the output of two runs of the MeRaLi system: in the first we only considered pairs where the system had enough information on both concepts involved in the comparison (named *selected data* in the following), whilst in the second one we also considered cases where no sufficient information was available in COVER for at least one of the concepts at hand (*full data* in the following). In the selected data run we only retain those pairs for which a vector description was found in COVER, and at least two shared dimensions were found to be filled. Satisfying all these constraints is, in our opinion, necessary in order to be able to justify on which bases two concepts are deemed similar or not. Table 3.4 shows the percentage of dropped pairs in each data set in the *selected data* condition. Conversely, in the *full data* condition we

Table 3.4: Percentage of dropped pairs for the *selected data* run of the MeRaLi system.

Dataset	Dropped pairs
MC	17%
RG	15%
WS-Sim	12%
SemEval2017	9%

Table 3.5: Spearman (ρ) and Pearson (r) correlations obtained over the four datasets.

System	RG		MC		WS-Sim		SemEval 2017	
	ρ	r	ρ	r	ρ	r	ρ	r
COVER (selected data)	0.82	0.88	0.89	**0.91**	0.69	0.70	0.68	0.67
COVER (full data)	0.76	0.81	0.74	0.79	0.61	0.60	0.65	0.63
NASARI$_{\text{embed}}$ [1]	0.88	**0.91**	0.83	**0.91**	0.68	0.68	0.68	0.68
ADW [2]	**0.92**	**0.91**	-	-	0.75	0.72	-	-
PPR [3]	0.83	-	**0.92**	-	-	-	-	-
ConceptNet Numberbatch [4]	-	-	-	-	**0.83**	-	-	-
Luminoso [5]	-	-	-	-	-	-	**0.72**	**0.74**
word2vec [6]	0.84	0.83	-	-	0.78	**0.76**	-	-

[1] (Camacho-Collados, Pilehvar, Collier, et al., 2017a; Camacho-Collados, Pilehvar, and Navigli, 2015b, 2016)
[2] (Pilehvar and Navigli, 2015)
[3] (Agirre et al., 2009)
[4] (Robyn Speer, Chin, and Havasi, 2017)
[5] (Robyn and Lowry-Duda, 2017)
[6] (Mikolov, K. Chen, et al., 2013)

considered all pairs. In particular, for pairs lacking at least one vector representation, or where less than two shared dimensions were filled, we assign a default similarity score of half the maximum of the similarity range. The rationale underlying these two runs is to try to fully assess the COVER resource, by also investigating to what extent the available information is helpful to conceptual similarity, irrespective of its current coverage.

3.2.3 *Results and Discussion*

Table 3.5 illustrates the results obtained by the MeRaLi system in the experimentation. Compared to the *selected data* run, the strongest competitors in literature obtained 10% higher ρ correlation on the RG data set (Pilehvar and Navigli, 2015) (3% on the MC data set (Agirre et al., 2009)); 14% on the WS-Sim data set (Robyn Speer, Chin, and Havasi, 2017). The distance from state of the art figures is reduced when testing on the SemEval 2017 data set, where we obtained a ρ correlation 4% lower than the Luminoso system (Robyn and Lowry-Duda, 2017). If we consider the *full data* run, our results are some points lower, with minimum (3%) loss w.r.t. the *selected data* run on the SemEval data set.

In order to discuss our results, we focus on the SemEval dataset, that is by far more complete (with 500 word pairs) and varied with respect to the other ones. In fact, it contains named entities and multiword expressions, and covers a wide range of domains.[6]

One major concern is the amount of missing information: as reported in Table 3.4, almost 10% of word pairs were dropped, as either lacking from COVER or due to the lack of shared information, which prevented us from computing the similarity. Missing concepts may be lacking in (at least one of) the resources upon which the COVER is built: including further resources may thus be helpful to overcome this limitation. Also, integrating further resources in COVER would be beneficial to add further concepts *per* dimension, and to fill more dimensions, so to expand the set of comparisons allowed by the resource.

[6] Namely, the 34 domains available in BabelDomains, http://lcl.uniroma1.it/babeldomains/.

Table 3.6: Spearman (ρ) and Pearson (r) correlations (and their harmonic mean) obtained by the MeRaLi system over the three subsets in the *full data* and *selected data* variants.

full data	# pairs	ρ	r	harm.mean
entire data	500	0.65	0.63	0.64
entity-concept	50	0.51	0.45	0.48
entity-entity	50	0.54	0.60	0.57
concept-concept	400	0.67	0.66	0.67

selected data	# pairs	ρ	r	harm.mean
entire data	452	0.68	0.67	0.67
entity-concept	36	0.61	0.60	0.60
entity-entity	31	0.70	0.75	0.72
concept-concept	385	0.68	0.67	0.67

A discussion of our results on this data set also involves a thorough analysis of the data set itself. The terms in the data set can be naturally arranged into three main classes, involving respectively concept-concept comparisons (400 word pairs), entity-entity comparisons (50 word pairs), and entity-concept pairs (50 word pairs).

So we have re-run the statistical tests to dissect our results according to the three individuated partitions of the data set; the partial results are reported in Table 3.6.

ENTITY-CONCEPT PAIRS. Comparisons involving a concept and an entity are somehow different from those involving only concepts. We individuated two further sub-classes: the pairs where the entity is instance of (that is, in relation InstanceOf with) the class indicated by the concept (e.g., ‘Gauss-scientist’, ‘Harry Potter-wizard’, ‘NATO-alliance’, *etc.*), and cases where the relations intervening between the two words at stake are not more specific than a general relatedness (e.g., 'Joule-spacecraft', 'Woody Allen-lens', 'islamophobia-ISIS', *etc.*). We then reran the MeRaLi system on the 50 entity-concept pairs (36 pairs in the *selected data* variant), and obtained overall 0.51 ρ correlation (thus significantly lower, than the general figures reported in Table 3.5). This datum can be complemented by comparing it with the corresponding result in the *selected data* variant: in this case, we obtained 0.61 ρ correlation. Interestingly enough, by focusing on the subset of elements linked by the InstanceOf relationship, we achieved a 0.79 ρ correlation.

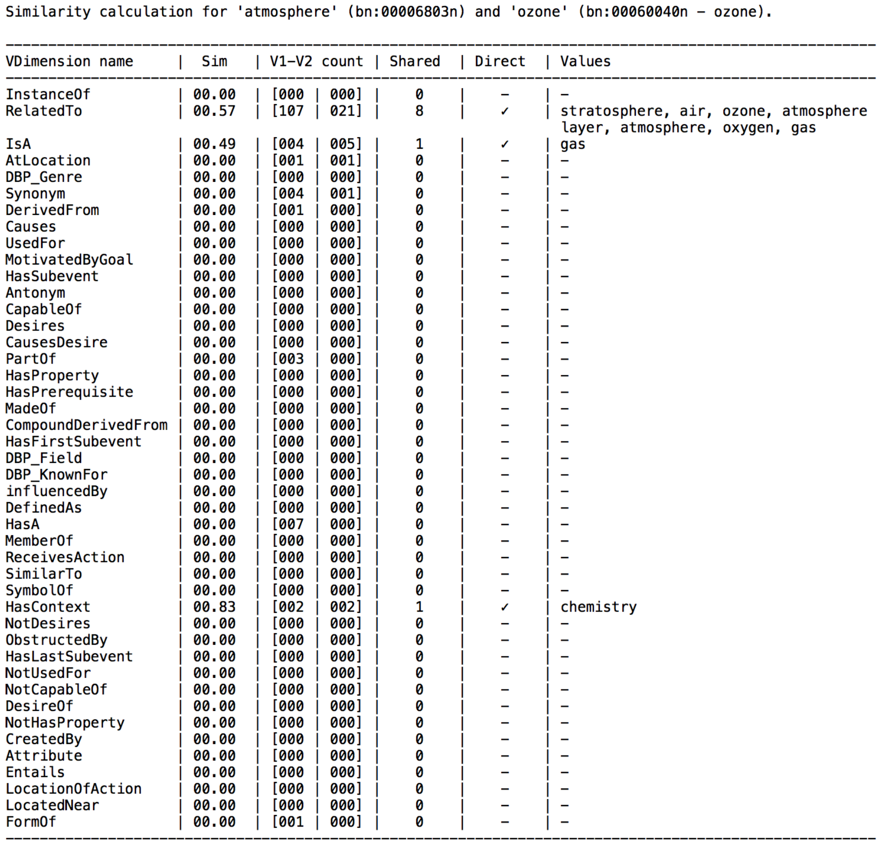
These results raise a question. Provided that the InstanceOf relationship is at the base of semantic similarity, COVER is appropriate to unveil semantic similarity for such pairs. However, in the remainder of the entity-concept pairs, the correlation with human judgments is still low. Even more, when the word pairs are not featured by the InstanceOf relationship, it is not simple to understand which sort of comparison is actually being carried out. From a cognitive perspective, it is difficult to follow the strategy adopted by human annotators in providing a similarity score for pairs such as 'Zara-leggings' (gold standard similarity judgment: 1.67 in a 0-4 scale, where 0 is dissimilar and 4 is the identity). In our approach, to assess the similarity between two elements entails individuating under which aspects they can be compared; it means to individuate a set of common properties and relations whose values can be directly compared. This explains that directly comparing a manufacturer and a product is nearly unfeasible, since their features can be hardly compared. In this case it is easy to grasp that the lack of shared (filled) dimensions between the entities may have determined many dropped pairs. Justifying the answer is perhaps helpful to give some information on the argumentative paths that can be possibly followed to assess semantic similarity. One major risk, in these respects, is that instead of *similarity*, the scores provided by human annotators rather refer to generic *relatedness*, which is generally acknowledged as a relation broader than similarity as illustrated in Section 2.3.2. Similar arguments also apply to meronyms. Let us consider, e.g., the pair 'tail-Boeing 747' (gold standard similarity judgment: 1.92): although each Boeing 747 has a tail, the whole plane (holonym) cannot be conceptually similar to its tail (meronym), in the same way a car is not similar to one of its wheels.

ENTITY-ENTITY PAIRS. As regards as the entity pairs, in the *selected data* experiment we obtained figures about 15% higher than in the *full data* condition: this is mainly due to the fact that some of the entities were not present in COVER (namely 31 pairs were used in the *selected data* condition vs. the 50 pairs in the *full data* condition). Conversely, the 70% agreement with human annotation is overall a reasonable performance, supporting the appropriateness of COVER. The absence of entities from COVER is easily explained: if either ConceptNet or BabelNet does not contain an element, then this is not present in COVER, that only hosts items that are present in both resources. In order to escape such limitation, next versions of COVER will contain information harvested also from further resources. The rate of agreement obtained experimenting with this subset of data closely approaches — limited to the *selected data* setting — the outstanding results obtained by the Luminoso team at the SemEval 2017 contest (Robyn and Lowry-Duda, 2017), and additionally benefits from the explanatory power allowed by the knowledge representation adopted in COVER.

CONCEPT-CONCEPT PAIRS. This is the principal class in the data set, counting 80% of word pairs in the *full data*, and 96% in the *selected data*. Although also items in this class pose some questions about the concepts at stake (such as comparisons between abstract and concrete entities like the pairs ‘coin-payment’, ‘pencil-story’ and ‘glacier-global warming’), our results over this subclass of data are by far less sensitive to the filtering performed in the *selected data* experiment (as it is illustrated in Table 3.6, the results of the MeRaLi system differ about 1% between the two settings). We interpret this result as one corroborating the claim that COVER is mature enough to ensure a reasonable coverage to compute conceptual similarity.

3.2.4 *Explaining Similarity: a* COVER *Speciality*

One of the most interesting perks of COVER consists in its ability to not only compute the similarity scores, but also to *natively* provide an explanation for them. Such feature is particularly interesting since often the score of similarity provided by a system can seem like an obscure number. It is difficult to demonstrate on which accounts two concepts are similar, especially if the score computation relies on complex networks or synthesised representations. However, thanks to the fact that COVER vectors contain explicit and human-readable knowledge, the explanation of the score is in this case allowed. Specifically, the COVER vectors adopted by the MeRaLi system provide human-readable features that



```
Similarity calculation for 'atmosphere' (bn:00006803n) and 'ozone' (bn:00060040n – ozone).

-------------------------------------------------------------------------------------------------------
VDimension name     |  Sim   | V1–V2 count | Shared  | Direct  | Values
-------------------------------------------------------------------------------------------------------
InstanceOf          | 00.00  | [000 | 000] |    0    |    –    | –
RelatedTo           | 00.57  | [107 | 021] |    8    |    ✓    | stratosphere, air, ozone, atmosphere
                                                                 layer, atmosphere, oxygen, gas
IsA                 | 00.49  | [004 | 005] |    1    |    ✓    | gas
AtLocation          | 00.00  | [001 | 001] |    0    |    –    | –
DBP_Genre           | 00.00  | [000 | 000] |    0    |    –    | –
Synonym             | 00.00  | [004 | 001] |    0    |    –    | –
DerivedFrom         | 00.00  | [001 | 000] |    0    |    –    | –
Causes              | 00.00  | [000 | 000] |    0    |    –    | –
UsedFor             | 00.00  | [000 | 000] |    0    |    –    | –
MotivatedByGoal     | 00.00  | [000 | 000] |    0    |    –    | –
HasSubevent         | 00.00  | [000 | 000] |    0    |    –    | –
Antonym             | 00.00  | [000 | 000] |    0    |    –    | –
CapableOf           | 00.00  | [000 | 000] |    0    |    –    | –
Desires             | 00.00  | [000 | 000] |    0    |    –    | –
CausesDesire        | 00.00  | [000 | 000] |    0    |    –    | –
PartOf              | 00.00  | [003 | 000] |    0    |    –    | –
HasProperty         | 00.00  | [000 | 000] |    0    |    –    | –
HasPrerequisite     | 00.00  | [000 | 000] |    0    |    –    | –
MadeOf              | 00.00  | [000 | 000] |    0    |    –    | –
CompoundDerivedFrom | 00.00  | [000 | 000] |    0    |    –    | –
HasFirstSubevent    | 00.00  | [000 | 000] |    0    |    –    | –
DBP_Field           | 00.00  | [000 | 000] |    0    |    –    | –
DBP_KnownFor        | 00.00  | [000 | 000] |    0    |    –    | –
influencedBy        | 00.00  | [000 | 000] |    0    |    –    | –
DefinedAs           | 00.00  | [000 | 000] |    0    |    –    | –
HasA                | 00.00  | [007 | 000] |    0    |    –    | –
MemberOf            | 00.00  | [000 | 000] |    0    |    –    | –
ReceivesAction      | 00.00  | [000 | 000] |    0    |    –    | –
SimilarTo           | 00.00  | [000 | 000] |    0    |    –    | –
SymbolOf            | 00.00  | [000 | 000] |    0    |    –    | –
HasContext          | 00.83  | [002 | 002] |    1    |    ✓    | chemistry
NotDesires          | 00.00  | [000 | 000] |    0    |    –    | –
ObstructedBy        | 00.00  | [000 | 000] |    0    |    –    | –
HasLastSubevent     | 00.00  | [000 | 000] |    0    |    –    | –
NotUsedFor          | 00.00  | [000 | 000] |    0    |    –    | –
NotCapableOf        | 00.00  | [000 | 000] |    0    |    –    | –
DesireOf            | 00.00  | [000 | 000] |    0    |    –    | –
NotHasProperty      | 00.00  | [000 | 000] |    0    |    –    | –
CreatedBy           | 00.00  | [000 | 000] |    0    |    –    | –
Attribute           | 00.00  | [000 | 000] |    0    |    –    | –
Entails             | 00.00  | [000 | 000] |    0    |    –    | –
LocationOfAction    | 00.00  | [000 | 000] |    0    |    –    | –
LocatedNear         | 00.00  | [000 | 000] |    0    |    –    | –
FormOf              | 00.00  | [001 | 000] |    0    |    –    | –
-------------------------------------------------------------------------------------------------------
```

Figure 3.6: Log of the comparison between the concepts *atmosphere* and *ozone* in MeRaLi. The 'V1-V2 count' column reports the number of concepts for a certain dimension in the first and second vector, respectively; the column 'Shared' indicates how many concepts are shared in the two conceptual descriptions along the same dimension; and the column 'Values' illustrates (the nominalization of) the concepts actually shared along that dimension.

are compared in order to obtain a similarity score. The explanation for this score can thus be obtained by simply reporting which values were a match in the two compared vectors. Ultimately, a simple Natural Language Generation approach has been devised on top of the score computation: a linguistic template is filled with the features in common between the two vectors, dimension by dimension. For instance, given the comparison between *atmosphere* and *ozone* in Figure 3.6 we can directly obtain the explanation:

> The similarity between *atmosphere* and *ozone* is 2.52 because they are *gas*; they share the same context *chemistry*; they are re-

```
The similarity between lizard and crocodile is 1.99 because
- they Are reptile;
- they are RelatedTo reptile, Caiman, fauna, diapsid.

The similarity between Harry Potter and wizard is 2.50 because
- they are RelatedTo spell, magic, magician, wand.

The similarity between beach and coast is 2.79 because
- they Are shore;
- they are semantically SimilarTo shore, formation;
- they are RelatedTo shore, coast, weather, seaboard, island, shell,
    wave.

The similarity between sodium chloride and salt is 3.56 because
- they are MadeOf sodium_chloride, ion, crystal;
- they can be found AtLocation Shaker_(laboratory), seawater, water,
    nutrient, mine, salt_mine;
- they Are binary_compound, taste, chemical_compound, Ionic_compound,
    spice, crystal, sodium_chloride, seasoning, inorganic_compound;
- they are UsedFor seasoning, nutrient;
- they share the same Context chemistry, inorganic_compound;
- they are SemanticallyOpposite of carbohydrate,
    Swedish_ethyl_acetate_method, vinegar;
- they are partOf seawater, sea;
- they are SimilarTo Sharp_(flour);
- they are SemanticallySimilarTo saltiness, sodium_chloride, salinity,
    salt;
- they are RelatedTo magnesium_lactate, Mevalonic_acid, cholic_acid,
    sulfate, halobacterium, benzoate, sulfonate, monosodium_glutamate,
    Glutaric_acid;
- they are DerivedFrom salinity, sodium, chloride, sodium_carbonate.
```

Figure 3.7: Some examples of the explanations that can be generated with the COVER resource.

> lated to *stratosphere*, *air*, *atmosphere*, *layer*, *ozone*, *atmosphere*, *oxygen*, *gas*.

by simply extracting the shared values among the two considered vectors. Other examples of explanations built following this approach can be found in Figure 3.7.

3.2.4.1 *Experimentation*

We also performed a pilot experimentation in order to evaluate the produced explanations (Colla, Mensa, Radicioni, and Lieto, 2018). In

Table 3.7: The pairs of terms employed in each questionnaire, referred to as Q1-Q4.

Q1	Q2
desert, dune	lizard, crocodile
palace, skyscraper	sculpture, statue
mojito, mohito	window, roof
city center, bus	agriculture, plant
beach, coast	flute, music
videogame, pc game	demon, angel
medal, trainers	income, quality of life
butterfly, rose	underwear, body
Wall Street, financial market	Boeing, plane
Apple, iPhone	Caesar, Julius Caesar

Q3	Q4
basilica, mosaic	car, bicycle
snowboard, skiing	democracy, monarchy
pesticide, pest	pointer, slide
level, score	flag, pole
snow, ice	lamp, genie
myth, satire	digit, number
sodium chloride, salt	coin, payment
coach, player	surfing, water sport
Zara, leggings	Harry Potter, wizard
Cold War, Soviet Union	Mercury, Jupiter

particular, we were interested in looking at the content provided as explanation rather than their linguistic realisation, which we kept very basic at this stage.

EXPERIMENTAL SETTING. The experiment was built by selecting 40 random pairs from the 'SemEval-2017 Task 2' dataset, the same adopted in the previous evaluation[7] (Table 3.7). Such pairs have been arranged into 4 questionnaires, that were administered to 33 volunteers, aged from 20 to 23. All recruited subjects were students from the Computer Science Department of the University of Turin (Italy); none of them was an English native speaker.

Questionnaires were split into 3 main sections:

[7] Actually the pair ⟨*mojito*,*mohito*⟩ was dropped in that 'mojito' was not recognised as a morphological variant of 'mohito' by most participants.

- in the *task* 1 we asked the participants to assign a similarity score to 10 term pairs (in this setting, scores are continuous in the range $[0, 4]$, as it is customary in the international shared tasks on conceptual similarity Camacho-Collados, Pilehvar, Collier, et al., 2017a);
- in the *task* 2 we asked them to explain in how far the two terms at stake were similar, and then to indicate a new similarity score (either the same or different) to the same 10 pairs as above;
- in the *task* 3 the subjects were given the automatically computed score along with the explanation built by our system. They were requested to evaluate the explanation by expressing a score in a $[0, 10]$ Likert scale, and also to provide some comments on missing/wrong arguments, collected as open text comments.

Each volunteer compiled one questionnaire (containing 10 term pairs), which on average took 20 minutes.

RESULTS AND DISCUSSION. The focus of the experimentation was the assessment of the quality of the automatically computed explanations (*task* 3): MeRaLi's explanations obtained, on average, the score of 6.62 (standard deviation: 1.92). Our explanations and the scores computed automatically have been overall judged to be reasonable.

By examining the 18 pairs that obtained an averaged poor score (≤ 6), we observe that either few information was available, or it was basically wrong. In the first case, we counted 12 pairs with only one or two shared concepts: almost always these explanations were evaluated with low scores (on average, 4.48). We found only one notable exception about the pair ⟨*Boeing*, *plane*⟩ whose explanation was

```
The similarity between Boeing and plane is 2.53 because they
are related to airplane, aircraft.
```

This explanation obtained an average score of 8.63. We hypothesise that this greater appreciation is due to the fact that even if only two justifications are provided, they match the most salient (based on common-sense accounts) traits between the two considered concepts. It would seem thus that the quality of a brief explanation heavily depends on the presence of those particular and meaningful traits. In the remaining 6 pairs, vice versa, there is enough though wrong information, possibly due to the selection of the wrong meaning for input terms. In either cases, we observe that the resource still needs being improved for what pertains its coverage and the quality of the hosted information (since it is automatically built by starting from BabelNet, it contains all noise therein).

The first and second task in the questionnaire can be thought of as providing evidence to support the result in the third one. In particular, the judgements provided by the volunteers closely approach the scores in the gold standard, as it is shown by the high (over 80%) Spearman's (ρ) and Person's (r) correlations (Table 3.8). The first two rows show the average agreement between the scores *before* producing an explanation for the score itself (Gold - avg scores (*task* 1)), and *after* providing an explanation (Gold - avg scores (*task* 2)). These figures show that even human judgement can benefit from producing explanations, as the scores in *task* 2 showcase a higher correlation with the gold standard scores. Additionally, the output of the system exhibits a limited though significantly higher correlation with the similarity scores provided after

Table 3.8: Correlation between the similarity scores provided by the subjects interviewed and the scores in the Gold standard. The bottom row shows the correlations between the scores gold standard and the scores computed by our system

	Spearman's ρ	Person's r
Gold - avg scores (task 1)	0.83	0.82
Gold - avg scores (task 2)	0.85	0.83
COVER - avg scores (task 1)	0.71	0.72
COVER - avg scores (task 2)	0.72	0.73
Gold - COVER	0.79	0.78

trying to explain the scores themselves (COVER - avg scores (*task* 1) condition *vs.* COVER - avg scores (*task* 2)).

In order to further assess our results we also performed a qualitative analysis on some spot cases. For the pair ⟨*Mercury, Jupiter*⟩ the MeRaLi system computed a semantic similarity score of 2.29 (the gold standard score was 3.17), while the average score indicated by the participants was 3.43 (*task* 1) and 3.29 (*task* 2). First of all, this datum corroborates our approach that computes the similarity between the closest possible senses (please refer to Equation 2.2): it never happened that any participant raised doubts on the meaning of Mercury (always intended as the planet), whilst *Mercury* can be also a metallic chemical element, a Roman god, the Marvel character who can turn herself into mercury, and several further entities.

The open text comments report explanations such as that Mercury and Jupiter are *'both planets, though different'*. In this case, the participants acknowledge that the two entities at stake are planets but rather different (e.g., the first one is the smallest planet in the Solar System, whilst the second one is the largest). The explanation provided by our system is:

> `The similarity between Mercury and Jupiter is 2.29 because they are` *`planet`*`; they share the same context` *`deity`*`; they are seman-`

> tically similar to *planet*; they are related to *planet*, *Roman_deity*, *Jupiter*, *deity*, *solar_System*.

In this case, our explanation received an average score of 9.57 out of 10. Interestingly enough, even though the participants indicated different similarity scores, they assigned a high quality score to our explanation, thus showing that it is basically reasonable.

As a second example we look at the pair ⟨*myth*, *satire*⟩. The similarity score and the related explanation of such terms are:

> The similarity between *myth* and *satire* is 0.46 because they are *aggregation*, *cosmos*, *cognitive_content*; they are semantically similar to *message*; they form *aggregation*, *division*, *message*, *cosmos*, *cognitive_content*.

In this case, the gold standard similarity value was 1.92, the average scores provided by the participants 1.57 (*task* 1) and 1.71 (*task* 2). Clearly, the explanation was not satisfactory, and it was rated 4.49 out of 10. The participants gave no clear explanation about their judgement (in *task* 2) nor informative comments/criticisms on the explanation above (in *task* 3). One possible reason behind the poor assessment might be found in the interpretation of the *satire* term. If we consider satire as the ancient literary genre where characters are ridiculed, the explanation becomes more coherent: they are forms of *aggregation* as it was for any sort of narrative in the ancient (mostly Latin) culture; they also both deliver some message, either explaining some natural or social phenomenon and typically involving supernatural beings (like myth), or criticising people's vices, particularly in the context of contemporary politics (like satire). This possible meaning has been considered only by 2 out of 8 participants, that mostly intended satire as a generic ironic sort of text. Even in this case, whilst the output of MeRaLi was rather unclear and questionable, the explanation shows some sort of coherence, although not immediately sensible for human judgement. In such cases, by resorting to an inverse engineering approach, the explanation can be used to figure out which senses (underlying the terms at hand) are actually intended.

3.3 USING COVER

Besides the word and concept similarity, COVER has been successfully employed in different tasks such as conceptual categorisation, keyword extraction and abstractness extraction. In the following section we will illustrate how COVER has been used as a key component to deal with this tasks.

3.3.1 COVER *and Conceptual Categorization*

A smaller and more specific version of COVER has been plugged into the Dual-PECCS system, a system devised to perform the conceptual categorization task by adopting an hybrid reasoning approach (Lieto, Minieri, et al., 2015; Lieto, Radicioni, and Rho, 2015, 2017; Lieto, Radicioni, Rho, and Mensa, 2017). The Dual-PECCS knowledge base puts together both vector representations and formal ontologies for the same conceptual entities (Figure 3.8): such information is then exploited by an hybrid reasoning algorithm that allows the resolution of simple riddles such as 'The animal that eats bananas' and 'The big mammal that eats plankton'. This system implements the dual process theory of reasoning and rationality which states that two different types of cognitive sys-

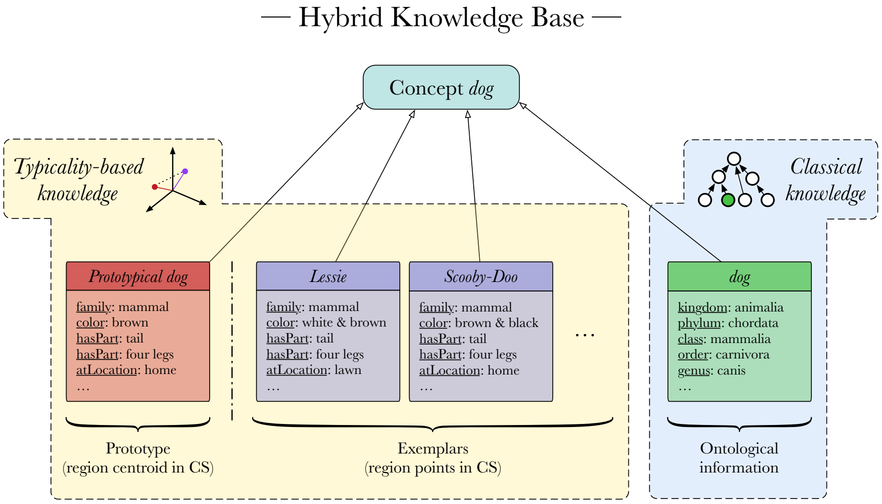


Figure 3.8: Heterogeneous representation of the *dog* concept in the hybrid knowledge base of DUAL-PECCS.

tems can coexist (Evans and Frankish, 2009; Kahneman, 2011). In this view, the systems of the first type (*type 1*) are phylogenetically older, unconscious, automatic, associative, parallel and fast. The systems of the second type (*type 2*) are more recent, conscious, sequential and slow, and featured by explicit rule following. *Type 1* processes have been designed to deal with prototypes- and exemplar-based retrieval, while *Type 2* processes have been designed to deal with deductive inference. COVER has been adopted to serve the *type 1* reasoning process, and to this end it has been rebuilt to represent only animals (as required by the dataset theme) and extended with animal-specific dimensions such as *family* and *color*. The categorization pipeline works as follows: a simple linguistic description such as 'the big striped feline that lives in the savanna' is provided to the system, which is expected to return the *tiger* category as answer. To do so, the system builds a vector filled with the details provided in the description (*feline*, *stripes*, *big*, and *savanna*) and computes the similarity between the query vector and all of the vectors inside the tipicality-based knowledge base (COVER). The result is a ranking of candidate entities that are then checked against the ontological KB: the best category is returned as result. Interestingly enough, we showed that common-sense descriptions such as that in the example cannot be easily dealt with with ontological inference alone, nor through other standard approaches (Lieto, Radicioni, and Rho, 2017; Lieto, Radicioni, Rho, and Mensa, 2017).

3.3.2 COVER *and Keyword Extraction*

COVER has also been employed for the resolution of the keywords extraction task (Colla, Mensa, and Radicioni, 2017). Our approach builds on the idea that in order to extract high quality keywords the semantic content of documents must be taken into consideration. We define the *relevance* of a keyword (a word in the document body) by means of its centrality w.r.t. the entities in the document title. In detail, the system starts from the lists $T = \{y_1, y_2, \ldots, y_L\}$ such that $y \in \textsf{document title}$ and $B = \{x_1, x_2, \ldots, x_M\}$ such that $x \in \textsf{document body}$, that contain the BabelNet synset IDs in the title and in the body of the document, respectively. We then compute the centrality c of the concepts corresponding to the terms x in the body as a function of their semantic relatedness to those in the title:

$$c(x) = \frac{1}{|T|} \sum_{y_i \in T} \text{semrel}(x, y_i). \tag{3.9}$$

We devised five metrics that implement the semrel function by exploiting different resources and techniques. Namely, we propose the following metrics: NASARI, NASARIE, UCI, UMASS and MeR-

ALi, that can be arranged into two classes of metrics: those based on NASARI conceptual representations, and those based on coherence measures.

Regardless of the employed metrics, for each document we select as the best keywords those with maximum centrality, that is:

$$\mathit{Keywords} = \operatorname*{argmax}_{x \in B} c(x).$$

USING NASARI VECTORS TO COMPUTE SEMANTIC RELATEDNESS. As our first measure, we exploit the semantic vectors of NASARI. In the following we will denote the concept identifier by y or x, and the corresponding vector by $\vec{y}$ or $\vec{x}$.

The semantic relatedness between a concept $x \in B$ and the concept $y \in T$ is computed by considering $\rho_x^{\vec{y}}$, that is the *rank* of x in the vector representation for y. More specifically, given two arbitrary elements x and y_i, we compute their relatedness as

$$\text{semrel}(x, y_i) = \left(1 - \frac{\rho_x^{\vec{y}_i}}{length(\vec{y}_i)}\right).$$

The rationale underlying this formula is that x is more relevant to the concept y_i if x has smaller rank (and heavier weight), i.e., x is found among the first concepts associated to y_i in $\vec{y}_i$. For example, if we inspect[8] the NASARI vector for the concept *door*, we find — in decreasing relevance order — that the third term associated to *door* is *window*, the tenth *wall*, the twelfth is *lock*, and around the hundredth

[8] For the sake of clarity in this example we consider the *lexical* rather than the *unified* vector, i.e. having terms in place of conceptual IDs that are actually used by the system.

position *interior door*: the above formula emphasizes the contribution of heavier features, having lower rank.

The centrality of the concept x with respect to each concept $y_i \in T$ can be determined as

$$\text{semrel}(x, y_i) = \begin{cases} 1 & \text{if } \rho_x^{\vec{y_i}} = 1; \\ 0 & \text{if } x \notin \vec{y_i}; \\ \left(1 - \frac{\rho_x^{\vec{y_i}}}{length(\vec{y_i})}\right) & \text{otherwise.} \end{cases}$$

Specifically, in case the concept x is found to have rank 1 for the concept y_i its relevance is supposed to be maximal to the meaning of y_i (it is likely the same term or a close term which is part of the same synset); conversely, in case it is not found in the vector associated to y (thus obtaining $\rho_x^{\vec{y_i}} = 0$), the relatedness (x, y_i) will not contribute anything to the overall centrality of x to the concepts in T.

USING NASARIE VECTORS TO COMPUTE SEMANTIC RELATEDNESS. We also explored the NASARIE version, that contains embedded vector representations of 300 dimensions; the computation of the centrality can be computed in this case by resorting to standard cosine similarity, thus

$$\text{semrel}(x, y_i) = cosSim(\vec{x}, \vec{y_i}).$$

USING UCI COHERENCE MEASURE TO COMPUTE SEMANTIC RELATEDNESS. Moreover, we propose two metrics, the UCI measure (Newman et al., 2010) and the UMass measure (Mimno et al., 2011) that — originally conceived for evaluating Latent Dirichlet Allocation

—, have been used in the automated semantic evaluation of different latent topic models (Stevens et al., 2012).[9]

Because both the UCI and the UMASS measures natively handle terms rather than concepts, after the semantic preprocessing phase, we need to translate back concepts into terms. However, by exploiting BabelNet, we map all synonyms for a given concept onto a single shared lexicalization, that is chosen as the most common term according to BabelNet counts. This strategy allows reconciling different terms underlying the same sense, thus preserving some semantic trait.

The UCI metrics (Newman et al., 2010) computes the cohesion between two terms w_1 and w_2 through their pointwise mutual information, that is

$$score(w_1, w_2, \epsilon) = \log \frac{p(w_1, w_2, \epsilon)}{p(w_1)p(w_2)},$$

where the probabilities are estimated by counting word co-occurrence frequencies in a sliding window over an external corpus, such as Wikipedia, Google or MEDLINE,[10] and the ϵ correction is used to ensure that the function always returns real numbers (presently ϵ is set to 1). In our setting, we are interested in computing the cohesion score between the terms in the body and the terms in the title, so that for each concept $x \in B$ lexicalized as w_x and $y_i \in T$ lexicalized as w_{y_i} we compute

$$\text{semrel}(x, y_i) = score(w_x, w_{y_i}, 1).$$

[9] In order to compute such measures we used the Palmetto library (Röder, Both, and Hinneburg, 2015).

[10] Specifically, in the Palmetto implementation, the pointwise mutual information (PMI) and word co-occurrence counts were computed by using Wikipedia as reference corpus (Röder, Both, and Hinneburg, 2015).

Using UMass coherence measure to compute semantic relatedness This metrics define a coherence score based on the co-occurrence of the terms w_1 and w_2 as (adapted from (Stevens et al., 2012))

$$score(w_1, w_2, \epsilon) = \log \frac{D(w_1, w_2) + \epsilon}{D(w_2)},$$

where $D(w_1, w_2)$ and $D(w_2)$ count the number of documents containing both w_1 and w_2, and only w_2, respectively. The adopted formula follows the rationale illustrated for the UCI metrics:

$$\text{semrel}(x, y_i) = score(w_x, w_{y_i}, 1),$$

where the concept $x \in B$ is lexicalized as w_x, and $y_i \in T$ is lexicalized as w_{y_i}.

Using COVER to compute semantic similarity. The last measure that we employed to compute the semantic relatedness is based on COVER. We rely on the MeRaLi system introduced in Section 3.2 to compute the semantic similarity between concepts in the title and those in the documents body according to the formula

$$\text{semrel}(x, y_i) = STRM(\vec{x}, \vec{y_i}),$$

where $\vec{x}$ and $\vec{y_i}$ represent the COVER vectors for the concepts $x \in B$ and $y_i \in T$, respectively and $STRM$ is the Symmetrical Tversky's Ratio Model (Equation 3.8).

3.3.2.1 *Evaluation*

In the last few years several sets of keywords-annotated documents have been collected, annotated and made available, that allow assessing algorithms and their underlying assumptions on scientific articles, news documents, Broadcast News and Tweets (see, for example, (Marujo et al., 2012)).

DATASET. We experimented on the Crowd500 dataset (Marujo et al., 2012), which has been extensively used for testing. The dataset contains overall 500 documents (450 for training and 50 for testing purposes), arranged into 10 classes: Art and Culture, Business, Crime, Fashion, Health, US politics, World politics, Science, Sport, Technology. Documents herein have been annotated by several annotators recruited through the Amazon's Mechanical Turk service. Each keyphrase is provided with a score equal to the number of annotators who selected it as a keyphrase.

PARTICIPANTS. In the following, for the sake of self-containedness, we report the experimental results obtained by (Jean-Louis et al., 2014), where the authors performed a systematic assessment of an array of keyword extractors and online semantic annotators. In particular, we report the results obtained by 2 keyword extractors that participated in the 'SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles' (namely, KP-Miner (El-Beltagy and Rafea, 2009) and Maui (Witten et al., 1999)), and 5 semantic annotators (AlchemyAPI, Zemanta, OpenCalais, TagMe, and TextRazor.[11]) With regards to Alchemy, both the keyword extraction (Alch Key) and concept tagging (Alch Con) services were considered. More details can be found in (Jean-Louis et al., 2014).

EXPERIMENTAL SETTING. We adopted the same setting as in (Jean-Louis et al., 2014), where two experiments have been carried out: in the first one the authors restricted to considering the top 15 keywords for each document in the dataset, while in the second one they considered all annotated keywords. Given the diversity of the metrics employed, some of them typically return a centrality score for each concept in the document (NASARIE, UCI, UMASS), while the other ones (NASARI and MeRaLi) are only able to express a centrality score for some of the concepts in the document. For this reason, we defined the number of keywords returned by each metrics by considering as minimum the number of keywords having positive centrality score, and as maximum the average of keywords provided for each document in the training set (this figure amounts to 48 keywords per document). Also, since all metrics assessed were used at a conceptual level, our output is mostly composed by individual keywords rather than by keyphrases: accordingly, in the evaluation of the results, we disregarded all keyphrases and focused on the keywords in the gold standard.

RESULTS. The Precision, Recall and F_1 score obtained obtained by testing on the Crowd500 dataset are illustrated in Table 3.9. The Precision indicates the percentage of correct keywords among those

[11] Available at the URLs http://www.alchemyapi.com/api/keyword-extraction/, http://developer.zemanta.com/, http://www.opencalais.com/, http://TagMe.di.unipi.it/ and http://www.textrazor.com/, respectively.

Table 3.9: Results obtained on the test set of the Crowd500 dataset: for each system Precision (P), Recall (R) and F_1 Score (F) are reported.

(a) Results on the top 15 keywords in the gold standard.

participant	*k*	P(%)	R(%)	F(%)
Alch Con	15	16.71	2.81	4.82
Alch Key	15	21.63	6.32	9.78
Calais_Soc	15	6.67	0.09	0.17
KP-Miner	15	**41.33**	8.05	13.48
Maui	15	35.87	9.78	15.37
TagMe	15	34.53	11.21	16.93
TxtRaz Top	15	15.78	5.02	7.62
Zem Key	15	29.75	5.15	8.78
NASARI	15	24.89	10.40	14.67
NASARIE	15	15.62	35.47	21.69
UCI	15	16.06	**44.40**	**23.59**
UMASS	15	15.49	42.53	22.71
MeRaLi	15	29.08	8.13	12.71

(b) Results on all keywords of the gold standard.

participant	*k*	P(%)	R(%)	F(%)
Alch Con	all	16.71	2.81	4.82
Alch Key	all	12.40	16.71	18.24
Calais_Soc	all	13.69	2.60	4.29
KP-Miner	all	40.19	14.46	21.27
Maui	all	27.46	20.30	23.34
TagMe	all	21.02	35.89	26.51
TxtRaz Top	all	6.28	11.52	8.13
Zem Key	all	29.75	5.15	8.78
NASARI	all	39.83	10.86	17.06
NASARIE	all	27.72	36.16	31.38
UCI	all	29.68	**46.28**	**36.17**
UMASS	all	26.76	43.08	33.02
MeRaLi	all	**50.36**	8.49	14.54

returned by the system, while the Recall is the percentage of correct keywords returned among all of the possible correct keywords, finally, F_1 is the harmonic mean between the two. Specifically, in Table 3.9(a) we present the results obtained by comparing the keywords extracted to the top 15 keywords in the Crowd500 dataset, while the results obtained by considering all of the gold standard keywords are provided in Table 3.9(b). Regarding the first experiment, over the top 15 keywords, we note that in 3 out of 5 of the considered metrics (namely, NASARIE, UCI and UMASS), the F_1 score is higher than those reported in the paper by (Jean-Louis et al., 2014). Also in the second experiment NASARIE, UCI and UMASS obtained highest F_1 score, whilst the results of NASARI and MeRaLi are featured by the highest precision.

DISCUSSION. Given the simplicity of the hypothesis being tested (that is: the title-body conceptual coherence is sufficient to individuate the keywords), the adopted metrics performed surprisingly well, and seem to confirm that our hypothesis is sound. We notice that in computing the results over the 15 top ranked keywords (Table 3.9(a)), the precision of all our measures is quite low, on average half of that obtained by KP-Miner, Maui and TagMe. In any case, this datum would make our metrics inapplicable in a real setting. Although the precision over all keywords (Table 3.9(b)) is in line with the other systems (except for KP-Miner, that has an advantage of around 10% on our score), the low precision over the first 15 keywords (that are the more relevant ones) shows that the ranking component in the extraction phase must be improved.

On the other side, one weakness of our experimentation (which is, admittedly, a preliminary one) is due to the fact that our results do not actually include keyphrases but only keywords, and thus they cannot be directly compared to those of the other systems. We started devising a module for the recognition of Named Entities (which is to date an open problem) to be integrated into the described system. However, even though we were forced to disregard keyphrases, at a closer inspection of the data, in some cases the annotated keyphrases seem to be rather inaccurate: for example, it is frequent to find locutions such as 'video below', 'although people', 'SeaWorld and', 'size allows' and many others.

Finally, by referring to Table 3.9(b) we note that the traditional trade-off between precision and recall seems to be intertwined with the degree of semantics adopted. In fact, the metrics based on MeRaLi— which is semantically more sophisticated than the other metrics and represents concepts as entities related to other concepts — obtained over 50% precision, whilst the UMASS metrics, which basically counts terms occurrence in documents, obtained 26.76% precision. A full account of

Table 3.11: Analysis of the Precision scores by domain (All-keywords experimentation).

Domain	NASARI	NASARIE	UCI	UMASS	MeRaLi
Tech	33.92	35.56	31.25	25.00	**60.00**
Sports	**34.05**	18.10	24.99	23.70	28.33
Business	40.29	30.76	27.08	27.50	**50.00**
US Politics	38.71	30.63	34.17	32.92	**66.67**
Art and Culture	**32.50**	21.95	23.75	22.08	20.00
Science	41.90	26.21	24.58	23.75	**59.58**
Health	33.81	20.39	27.08	22.92	**46.67**
World politics	**68.00**	41.95	46.44	46.44	34.00
Crime	45.12	27.92	27.08	21.25	**60.00**
Fashion	30.04	23.75	30.44	22.08	**78.33**
Median	39.83	27.72	29.68	26.76	50.36
Average	36.38	27.07	27.08	23.73	54.79
STDEV	10.96	7.30	6.73	7.72	18.28

the precision over the 10 domains is provided in Table 3.11: consistently with previous observations and findings, metrics with highest results have higher standard deviation: this fact is trivially explained by the fact that metrics that perform poorly get low scores on most of the domains, which tend to increase their stability (Jean-Louis et al., 2014).

Moreover, in Table 3.12 we present the number of keywords available on average over the 10 domains, and the actual number of keywords extracted through the considered metrics. These figures have been obtained in the experiment considering all keywords. By comparing the number of keywords returned by MeRaLi and NASARI, we observe that even in cases when MeRaLi returns ‘many’ keywords, its precision still scores high: this is the case, for example, of the domains Sports, Science, Crime and Fashion.

Table 3.12: Comparison between the average number of keywords actually returned by each metrics, and (first column) the average number of keywords available in the test set.

Domain	DS	NASARI	NASARIE	UCI	UMASS	MeRaLi
Tech	45	14	43	48	48	2
Sports	26	12	43	45	45	11
Business	37	10	45	48	48	2
US Politics	19	5	27	38	38	1
Art and Culture	21	5	39	48	48	1
Science	40	20	47	48	48	12
Health	33	14	44	48	48	3
World politics	18	3	20	34	34	9
Crime	37	5	48	48	48	11
Fashion	55	12	48	48	48	11

3.3.3 COVER *and Abstractness*

The common-sense provided by COVER has proven to be beneficial for the computation of the *abstractness* of concepts. Specifically, we extended COVER with abstractness scores by producing the Abs-COVER resource (Mensa, Porporato, and Radicioni, 2018a), then we exploited this annotations on nouns to produce abstractness scores on verbs (Colla, Mensa, Porporato, et al., 2018) and in parallel we exploited this extended version to tackle the metaphor detection task (Mensa, Porporato, and Radicioni, 2018b).

3.3.3.1 *Why Abstractness?*

We decided to focus on the aspect of concepts abstractness since ordinary experience shows that semantic representation, lexical access and processing of concepts can be affected by concepts' concrete/abstract status. Concrete meanings, closely related to the perceptual experience, are acknowledged to be more quickly and easily delivered in human communication than abstract meanings (Bambini, Resta, and Grimaldi, 2014). Such kind of information grasps a complex combination of experiential (e.g., sensory, motor) and strictly linguistic features, such as verbal associations arising through co-occurrence patterns and syntactic information (Vigliocco et al., 2009). Our intuition is that common-sense like information can be beneficial for the computation of concepts abstractness, thus COVER could constitute a potential starting point to successfully encode this type of knowledge. Information on conceptual abstractness impacts on many diverse NLP tasks, such as the word sense disambiguation task (O. Y. Kwong, 2008), the semantic processing of figurative language (Birke and Sarkar, 2006; Neuman et al., 2013), the automatic translation and simplification (Z. Zhu, Bernhard, and Gurevych, 2010), the characterisation of web queries with difficulty scores (Xing, Zhang, and Han, 2010), the processing of social tagging information (Benz et al., 2011), and many others, as well.

WHAT IS ABSTRACTNESS? The first issue consists in selecting a definition of abstractness (Iliev and Axelrod, 2017), since the term 'abstract' has two main interpretations: *i)* what is far from perception (as opposed to perceptible directly through the senses), and *ii)* what is more general (as opposed to low-level, specific). To implement the second view, the concreteness or *specificity* —the opposite of abstractness— can be defined as a function of the distance intervening between a concept and a parent of that concept in the top-level of a taxonomy or ontology (Changizi, 2008). This definition could then be used to easily compute abstractness on any given ontology-like resource (like WordNet or BabelNet) without any additional information from human beings.

On the other side, the first definition appears to better correlate with the human notion of 'abstract' (Theijssen et al., 2011).

In the process of extending COVER we refer to the first definition of abstractness, since the resource appears to be able to grasp the key aspects beneficial to the computation of how much a concept is perceivable or rather abstract. Furthermore, a novel aspect of the work consist in considering abstractness as a feature of word meanings (concepts) rather than as a feature of word forms (terms).

As a result, we propose Abs-COVER,[12] which enriches all concepts in COVER by providing an abstractness score ranging in the $[0,1]$ interval, where 0.0 indicates fully concrete concepts, and 1.0 stands for maximally abstract concept.

3.3.3.2 *Building* Abs-COVER

The annotation of COVER follows the simple principle that any entity which is a child of the concept *physical entity* in the hierarchy of WordNet,[13] can be considered concrete, while if it does not it can be considered abstract. The algorithm consists of two steps: the first aims at providing every concept with a base abstractness score, which is then refined in the latter step.

STEP 1: THE BASE SCORE. We take in consideration every *entity* e (i.e., concept defined via its BabelNet synset ID) in COVER which could be either a value assigned to a certain dimension or have a vector representation itself. We then execute Algorithm 1 in order to assign a base score to it:

[12] The download link for Abs-COVER can be found in Appendix B.

[13] The synset for *physical entity* has ID `wn:00001930n`.

Algorithm 1: Step 1 function.

```
   Input: a BabelNet synset $e$
   Output: the base abstractness score of the COVER element
           corresponding to $e$
   Function BaseScore(e):
1      $S \longleftarrow$ WordNetHypernyms($e$)
2      if $S \neq \emptyset$ then
3          if physical entity $\in S$ then
               return 0
           else
               return 1
       else
           $H \longleftarrow$ BabelNetHypernyms($e$)
4          $W \longleftarrow \bigcup_{h \in H}$ WordNetHypernyms($h$)
5          if $W \neq \emptyset$ then
6              if physical entity $\in W$ then
                   return 0
               else
                   return 1
           else
7              $g \longleftarrow$ GetMainBabelNetGloss($e$)
8              $N \longleftarrow$ Babelfy($g$)
               $G \longleftarrow []$
               for each $n$ noun concept $\in N$ do
9                  $q \longleftarrow$ GetGlossConceptAbstractness(n)
10                 if $q \geq 0$ then
                       append $q$ to $G$
11             if $G$ is not empty then
                   return average of scores in $G$
               else
12                 return $-1$
```

(a) We access BabelNet to obtain the list of WordNet synset IDs associated to e. We then collect the hypernyms set of this concepts in WordNet: if this set contains *physical entity* then we assign a base score of 0; otherwise it is set to 1 (Algorithm 1, lines 3);

(b) if (a) fails (i.e., no WordNet synset ID can be found for e), we collect instead its BabelNet hypernyms and once again search for them in WordNet: if at least one of e hypernyms has *physical entity* among its hypernyms, the base abstractness score of e is set to 0, and to 1 otherwise (6);

(c) if (b) fails (i.e., e has no hypernyms in BabelNet or none of them has an associated WordNet synset ID), we retrieve from BabelNet the main gloss for e (Algorithm 1, lines 7), disambiguate it,[14] thus obtaining a new set of concepts describing the entity. The steps (a) and (b) are then performed on each of gloss concepts, and all the valid scores are finally averaged to compute the base score of e (Algorithm 1, lines 9–11).

(d) If all of the above steps fail we assign the value -1, indicating that no suitable score could be computed (Algorithm 1, lines 12).

STEP 2: TUNING THE SCORES. Once we have associated a base score to every entity e with a base score, we can smooth it by employing the common-sense knowledge encoded in COVER itself (Algorithm 2). Given a vector $\vec{c}$ in the resource, we take into consideration a subset of its dimensions:[15] all the base abstractness scores assigned to the concepts filling these dimensions are retrieved and averaged as the $s_{\text{values-avg}}$ score. Concepts having an invalid score are discarded (Algorithm 2, lines 1 and 2). The score $s_{\text{values-avg}}$ is then in turn averaged with $s_{\text{vec-base}}$, that is the base score of $\vec{c}$ (Algorithm 2, line 3), thus obtaining the final score for the COVER vector. If either $s_{\text{vec-base}}$ or $s_{\text{values-avg}}$ are invalid scores, the final score of $\vec{c}$ is set to the only valid score available.

To ensure that the order in which the COVER vectors are considered during this step does not impact on the results of the computation, we do not dynamically update the abstractness scores but we rather considered the base scores as 'frozen' for the whole process. Moreover, the tuning

[14] The disambiguation is performed by using Babelfy APIs (http://babelfy.org/).

[15] We presently consider the following dimensions: RelatedTo, FormOf, IsA, Synonym, DerivedFrom, SimilarTo and AtLocation.

Algorithm 2: Step 2 function.

```
Input: a COVER element $elem$, a set of COVER dimensions $D$, a set
       $A$ of pairs $(c, a)$, with $c$ COVER element and $a$ base
       abstractness score of $c$
Output: the refined abstractness score for $elem$
Function TuneScores($elem, D, A$):
    $s_{\text{vec-base}} \longleftarrow A(elem)$          // find the score of v in A
    if $elem$ is a COVER vector then
        $L \longleftarrow []$
        for each dimension $dim \in D$ do
            for each value $v \in elem.dim$ do
                $abstr_v \longleftarrow A(v)$
                if $abstr_v \geq 0$ then
                    append $abstr_v$ to $L$
        if $L$ is not empty then
            $s_{\text{values-avg}} \longleftarrow$ average of scores in $L$
        else
            $s_{\text{values-avg}} \longleftarrow -1$
        case $s_{vec\text{-}base} \geq 0$ AND $s_{values\text{-}avg} \geq 0$ do return $\dfrac{s_{\text{vec-base}} + s_{\text{values-avg}}}{2}$
        case $s_{values\text{-}avg} \geq 0$ do return $s_{\text{values-avg}}$
        otherwise do return $s_{\text{vec-base}}$
    else
        return $s_{\text{vec-base}}$
```

algorithm was applied only once to avoid a potential drift from the precise base scores obtained via WordNet. In the end, any concept that is a child of the *physical entity* in WordNet retains an abstractness score lesser than or equal to 0.5. The final distribution of abstract and concrete vectors can be found in Figure 3.9: each line illustrates the amount of Abs-COVER vectors that fall in a specific abstractness score range. We can observe that concrete and abstract vectors are well separated, and in particular, the average score of concrete vectors (i.e., having score lower than or equal to 0.5) is 0.153 and the average score of abstract vectors (i.e, with score greater than 0.5) is 0.837. Table 3.13 reports instead the average abstractness of the 15 most populate dimensions of Abs-COVER for concrete and abstract concepts. For instance, we can observe that the IsA dimension averages at 0.215 for concrete vectors, and 0.787 for abstract vectors.

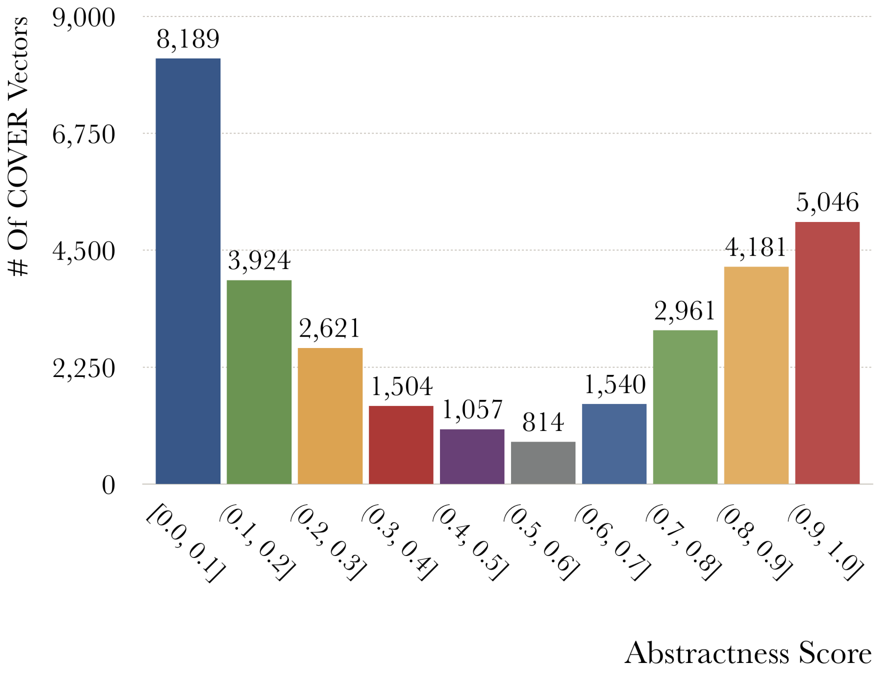


Figure 3.9: Distribution of COVER vectors by abstractness score.

These figures show that the annotation was coherent and thus qualitatively corroborate the proposed approach, however, to better estimate the quality of Abs-COVER we also designed an experimentation to study the correlation between its scores and human judgements. To this end we relied upon two datasets: the Medical Research Council Psycholinguistic Dataset (Coltheart, 1981) and the Brysbaert Dataset (Brysbaert, Warriner, and Kuperman, 2014). Since these datasets provide *word* abstractness rather then *sense* abstractness we implemented five disambiguation strategies to select the Abs-COVER vectors to be considered, and we additionally performed a pilot experimentation in which we manually disambiguated 150 pairs of words. The obtained figures are either in line or directly improve on state of the art approaches, such as the ones by Xing, Zhang, and Han (2010) and by Theijssen et al. (2011), showing that the annotation of COVER produced a reliable and competitive resource (Mensa, Porporato, and Radicioni, 2018a). A

Table 3.13: Average abstractness score in COVER vectors' dimensions. Starred dimensions indicate those actually used in the second step.

Dimension	Average Abstractness	
	Concrete Concepts	Abstract Concepts
RelatedTo*	0.293	0.694
IsA*	0.215	0.787
Synonym*	0.254	0.772
HasContext	0.632	0.805
FormOf*	0.127	0.777
DerivedFrom*	0.227	0.736
Antonym	0.312	0.750
AtLocation*	0.261	0.537
HasA	0.150	0.682
PartOf	0.181	0.681
SimilarTo*	0.241	0.751
UsedFor	0.464	0.719
HasProperty	0.385	0.727
Cause	0.450	0.811
CapableOf	0.473	0.687
HasPrerequisite	0.339	0.723

future development of the annotation algorithm could consist in the introduction of a hyper parameter to account for the weighting of the average between $s_{\text{vec-base}}$ and $s_{\text{values-avg}}$ in the tuning step. Being able to fine tune the impact of the base value of abstractness against the average abstractness of the vector could in fact possibly improve the final abstractness scores.

3.3.3.3 *Annotating Verbs Abstractness*

Since Abs-COVER provides conceptual representations and abstractness scores for nouns, we also explored the hypothesis that noun abstractness could be exploited to obtain verb abstractness (Colla, Mensa, Porporato, et al., 2018). We represent the meaning of verbs in terms of their argument distribution, by following the intuitive notion that abstract verbs are expected to have more abstract dependents than concrete ones. For example, let us consider the verb *drop*. To drop may be —concretely— intended as "to fall vertically". In this case, it takes concrete nouns as dependents, such as, e.g., in "the bombs are dropping on enemy targets". In a more abstract meaning to drop is "to stop pursuing or acting": in this case its dependents are more abstract nouns, such as, e.g., in "to drop a lawsuit". Although some counterexamples may also be provided, we found that this assumption holds in most cases.

Once again, we made use of the COCA dataset[16] to retrieve the 1,000 most common verbs and then we collected their dependents by sampling 3,000 occurrences of such verbs in the WaCkypedia_EN *corpus*, a 2009 dump of the English Wikipedia, containing about 800 million tokens, tagged with POS, lemma and full dependency parsing (Baroni, Bernardini, et al., 2009).[17] All trees containing the verbs along with their dependencies were collected and disambiguated via Babelfy. We retained all verb senses with at least 5 dependents that are present in Abs-COVER. The abstractness score of each sense has been computed by averaging the abstractness scores of all its dependents.

EVALUATION. The evaluation on the verb scores was performed on the 5,369 verbs of the Brysbaert Dataset (Brysbaert, Warriner, and Kuperman, 2014). A key issue consisted in finding the correct disambiguation for each verb in the dataset. To this aim we developed four different disambiguation strategies:

[16] http://corpus.byu.edu/full-text/.
[17] http://wacky.sslmit.unibo.it/doku.php?id=corpora.

	MaxAbs	*MinAbs*	*MaxDep*	*BestSns*
Pearson r	0.4163	0.4581	0.5103	0.4729
Spearman ρ	0.4037	0.4690	0.5117	0.4792

Table 3.14: Correlation results obtained by comparing our system's abstractness scores against the human ratings in the Brysbaert Dataset.

1. the sense with highest abstractness (*MaxAbs*);
2. the sense with lowest abstractness (*MinAbs*);
3. the sense with the highest number of dependents (*MaxDep*);
4. the sense returned as the best sense through the BabelNet API (*BestSns*).

The obtained correlations between the Brysbaert annotations and our annotations are reported in Table 3.14.

The differences shown in table provide tangible evidence that the problem of selecting the correct sense for a verb is a crucial one. E.g., if we consider the verb 'eat', the sense described as "Cause to deteriorate due to the action of water, air, or an acid (example: The acid *corroded* the metal)" and the sense described as "Worry or cause anxiety in a persistent way (What's *eating* you?)" exhibit very different abstractness scores. In order to decouple the assessment of the abstractness scores from that of the sense selection, we also randomly selected 400 verbs, and manually annotated them with an *a priori* reasonable sense.[18] This annotation process is definitely an arbitrary one (only one annotator, thus no inter annotator agreement was recorded, *etc.*), and it should be considered as an approximation to the senses underlying the human ratings available in the Brysbaert Dataset. The correlation scores significantly raise, as illustrated in the first column of Table 3.15, thus confirming the centrality of the sense selection step.

[18] Disambiguation proper would require to select a sense in accordance with a given context.

	FULL-400	Pruning ϑ_1
Pearson r	0.6419	0.6848
Spearman ρ	0.6634	0.6854

Table 3.15: Correlation scores obtained by manually choosing the main sense for 400 verbs (column FULL-400), and correlation scores obtained by removing from the FULL-400 verbs those with abstractness $\leq .1$ (column ϑ_1 pruning).

Furthermore, we observed that most mismatches in the computation of the abstractness scores occur when the verb is featured by very low (lower than 0.1) abstractness score. To corroborate such intuition, we have then pruned from our data set the verbs whose annotated score is lower than a threshold $\vartheta_1 = 0.1$, finally yielding 383 verbs. In this experimental setting we obtained higher correlation scores, thereby confirming that the computation of more concrete entities needs to be improved, as illustrated in the second column of Table 3.15.

The obtained results point out an increased difficulty in determining the scores of concrete verbs and the relevance of the disambiguation step, which was expected since verbs are known to be more polysemous then nouns.

3.3.3.4 *Abstractness and Metaphor Detection*

As a final application for Abs-COVER we developed a preliminary experimentation on the metaphor detection task (Mensa, Porporato, and Radicioni, 2018b), which represents to date an extraordinary challenge for computational linguistics. Dealing with metaphors has relevant impact on our ability to build agents and systems that understand Natural Language and text documents: annotating metaphoric constructions by linking the metaphor elements to existing resources is a crucial step to make text documents more easily accessible by machines.

METAPHOR CATEGORIZATION. Provided that different categorizations of metaphors can be drawn, we refer to the threefold (not exhaustive) categorization of metaphors proposed in (Krishnakumaran and X. Zhu, 2007). In this view, *Type I* metaphors are in the form "*smb/sth* is *sth*" (e.g., "He is a monster"), in which something or somebody is said to be of a kind that is not correct in a literal sense; *Type II* metaphors are in the form "*smb/sth verb sth*" (e.g., "I shot down all his arguments"), where an action is performed by or on something that cannot properly perform an action of that sort; *Type III* metaphors are in the form "*adj noun*" (e.g., "A brilliant idea"), where an adjective is associated to a concept that cannot have the quality expressed in a literal sense. In our introductory work we focused on metaphors of Types I and II, and disregarded those of any different type.

METAPHOR DETECTION. The system works as follows: given a sentence $\mathcal{S}$ along with its parse tree $\triangle(\mathcal{S})$, we individuate the dependency patterns corresponding to Type I and II metaphors, which we denote as $\overline{\wedge}(\mathcal{S}) \subset \triangle(\mathcal{S})$.

In order to do so, we preprocess the sentence by parsing and disambiguating it. Namely, given in input the sentence $\mathcal{S} = \{t_1, t_2, \dots, t_n\}$ composed of n input terms, we parse it and obtain the parse tree $\triangle(\mathcal{S})$; we then perform the word sense disambiguation of the terms in $\mathcal{S}$,[19] thus obtaining the set of concepts $\mathcal{C}(\mathcal{S}) = \bigcup_{i=1}^{n} \text{WSD}(t_i)$.

The metaphor detection algorithm consists then of the following steps:

1. Given the dependency patterns $\overline{\wedge}(\mathcal{S}) \subset \triangle(\mathcal{S})$ on the parse tree, we retain the corresponding concepts, $\mathcal{C}' = \bigcup_{\mathcal{C}} \overline{\wedge}(\mathcal{S})$; among

[19] We presently used Babelfy for the WSD and the Stanford CoreNLP, https://goo.gl/yxcRPF as our parser.

concepts $c' \in \mathcal{C}'$, we select *target* (that is, *subj* in Type I metaphors and a verb dependant in Type II metaphors) and *source* (*dir-obj* in Type I metaphors, and *verb* in Type II metaphors) of the metaphorical expression;

2. We label the sense as a metaphor if the target concept is more abstract than the source concept.

As an example, let us consider the sentence "*the past is a captor*". Given its parse tree, it can be recognized as a Type I metaphor in which the target is "*past*" and the source is "*captor*". After the WSD step, we access the vectors in Abs-COVER corresponding to the two concepts and discover that *past* has an abstractness score of 0.96 while the score for *captor* is 0.40: since the target concept is more abstract than the source concept, we can label the sentence as metaphorical.

EVALUATION. The pilot experimentation was performed on a portion of the Master Metaphors List (MML), a set of metaphors compiled by Lakoff and others in the '80s (Lakoff, Espenson, and A. Schwartz, 1991) and containing 1,728 sentences, each featured by at least one metaphor. From this set we extracted 75 sentences, 40 of which containing a metaphor of Type I, and 35 containing a metaphor of Type II. We then collected 75 additional non metaphoric sentences; syntactic constructions similar to those characterizing sentences with Type I and Type II metaphors were preserved.[20]

The system obtained a Recall of 0.70 and 0.74 on Type I and Type II, respectively, and a Precision of 0.56 (Type I) and 0.77 (Type II). The higher accuracy on Type II metaphors corroborates our hypothesis,

[20] The data set download link can be found in Appendix B.

thereby showing that for such (simpler) cases the comparison between target and source abstractness works fine. An explanation for the lower figures on Type I may stem from the fact that some of those metaphors require *projecting* features from the source onto the target (e.g., *lawyers are sharks*). In such cases, we conjecture that just considering the abstractness of the involved terms does not suffice since the metaphor is best recognized by projecting the features of ferocity and dangerousness — which is proper to sharks— onto lawyers. Remarkably, these are typically common-sense traits and so a future development could revolve around the exploitation of the structure of ABS-COVER (and not only its scores) to aid the algorithm.

4

LESSLEX

LessLex (Linking multilingual Embeddings to SenSe representations of Lexical items) is the second resource that we developed, consisting of a set of distributional vectors built by merging BabelNet and ConceptNet Numberbatch (Colla, Mensa, and Radicioni, 2020). Once again we show that the adoption of a sense layer can be beneficial to the resolution of the conceptual/word similarity task as well as two other downstream tasks: the contextual similarity and text similarity tasks.

4.1 BUILDING LESSLEX

The algorithm for the generation of LessLex is based on an intuitive idea: to exploit multilingual terminological representations in order to build precise and punctual conceptual representations. In doing so we started from CNN word embeddings and we build new sense embeddings by relying on the BabelNet sense inventory. We chose ConceptNet Numberbatch word embeddings (CNN from now on) as our starting point for a number of reasons: its vectors are to date highly accurate; all such vectors are mapped onto a single shared multilingual semantic space spanning over 78 different languages; it ensures reasonable coverage for general purposes use (Robyn Speer and Lowry-Duda, 2017); also, it allows dealing in a uniform way with multi-word expressions, compound words (Havasi, Robyn Speer, and Alonso, 2007), and even flexed forms; finally it is released under the permissive MIT License. Without loss of generality, we introduce our methodology by referring to nominal senses, while the whole procedure also applies to verb and adjectival senses, so that in the following we will switch between sense and concept as appropriated.

Each concept in LessLex is represented by a vector generated by averaging a set of CNN vectors. Given the concept c, we retrieve it in BabelNet to obtain the sets $\{\mathcal{T}^{l_1}(c), \ldots, \mathcal{T}^{l_n}(c)\}$ where each $\mathcal{T}^{l}(c)$ is the set of lexicalizations in the language l for c.[1] We then furtherly enrich this sets by extracting other terms from the concepts' English gloss and English Wikipedia Page Title (WT from now on) where available. The final result is the set $\mathcal{T}^{+}(c)$ that merges all the multilingual terms in each $\mathcal{T}^{l}(c)$ plus the terms extracted from the English gloss and WT. Only those terms that can be actually found in CNN are retained in $\mathcal{T}^{+}(c)$, so that the LessLex vector $\vec{c}$ can be finally computed by averaging all the CNN vectors associated to the terms in $\mathcal{T}^{+}(c)$.

4.1.1 *Selecting the Sense Inventory: Seed Terms*

The algorithm that generates LessLex takes in input a set of terms and generates a vector for each of their meaning. These *seed terms* are taken from different languages and different POS (nouns, verbs and adjectives are presently considered), and their meanings (retrieved via BabelNet) constitute the set of senses described by LessLex vectors.

[1] We presently consider the following languages: English (eng), French (fra), German (deu), Italian (ita), Farsi (fas), Spanish (spa), Portuguese (por), Basque (eus) and Russian (rus).

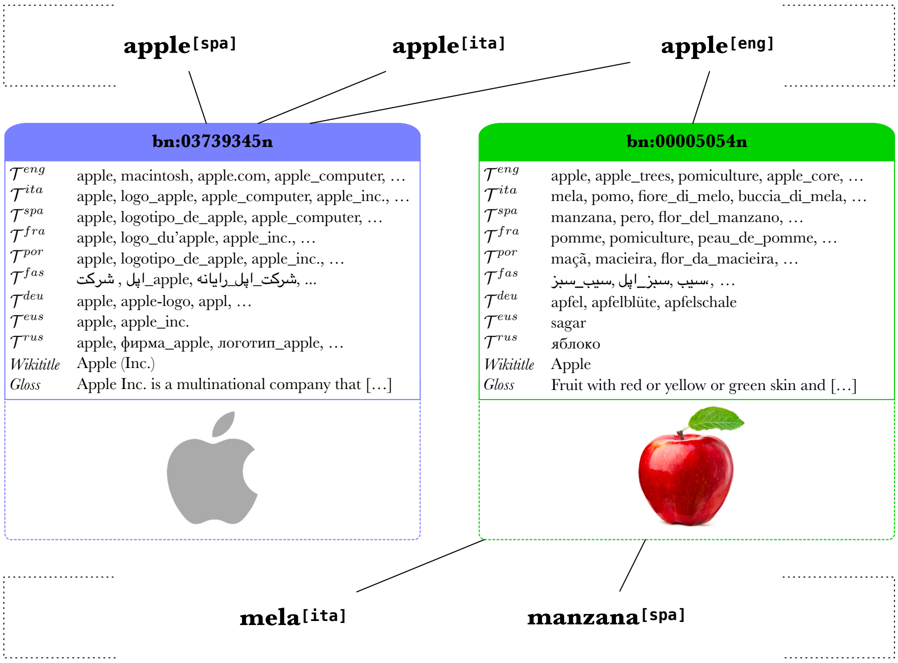

apple[spa]
apple[ita]
apple[eng]
bn:03739345n
$\mathcal{T}^{eng}$ apple, macintosh, apple.com, apple_computer, …
$\mathcal{T}^{ita}$ apple, logo_apple, apple_computer, apple_inc., …
$\mathcal{T}^{spa}$ apple, logotipo_de_apple, apple_computer, …
$\mathcal{T}^{fra}$ apple, logo_du'apple, apple_inc., …
$\mathcal{T}^{por}$ apple, logotipo_de_apple, apple_inc., …
$\mathcal{T}^{fas}$ شرکت , اپل_apple, شرکت_اپل_رایانه, …
$\mathcal{T}^{deu}$ apple, apple-logo, appl, …
$\mathcal{T}^{eus}$ apple, apple_inc.
$\mathcal{T}^{rus}$ apple, фирма_apple, логотип_apple, …
Wikititle Apple (Inc.)
Gloss Apple Inc. is a multinational company that […]
bn:00005054n
$\mathcal{T}^{eng}$ apple, apple_trees, pomiculture, apple_core, …
$\mathcal{T}^{ita}$ mela, pomo, fiore_di_melo, buccia_di_mela, …
$\mathcal{T}^{spa}$ manzana, pero, flor_del_manzano, …
$\mathcal{T}^{fra}$ pomme, pomiculture, peau_de_pomme, …
$\mathcal{T}^{por}$ maçã, macieira, flor_da_macieira, …
$\mathcal{T}^{fas}$ سیب_سبز, سبز_اپل, سیب, …
$\mathcal{T}^{deu}$ apfel, apfelblüte, apfelschale
$\mathcal{T}^{eus}$ sagar
$\mathcal{T}^{rus}$ яблоко
Wikititle Apple
Gloss Fruit with red or yellow or green skin and […]
mela[ita]
manzana[spa]

Figure 4.1: Retrieval of two senses for five seed terms in three different languages.

Naturally, due to the polysemy of language and to the fact that the seed terms are multilingual, different seed terms can retrieve the same meaning. It is important to note that seed terms do not affect the generation of a vector per se, but they rather determine the coverage of LessLex, since they are used to acquire the set of concepts that will be part of the final resource. Figure 4.1 illustrates this process for a few seed terms in English, Spanish and Italian. These terms provide two senses in total: `bn:03739345n` – *Apple (Inc.)* and `bn:00005054n` – *Apple (fruit)*. The first one is the meaning for $apple^{spa}$, $apple^{ita}$ and $apple^{eng}$, while the second one is a meaning for $manzana^{spa}$, $mela^{ita}$ and, again, $apple^{eng}$. Each synset contains all the lexicalizations in all languages, together with the English gloss and the WT. This information will be exploited for building $\mathcal{T}^{+}(c_{bn:03739345n})$ and $\mathcal{T}^{+}(c_{bn:00005054n})$ during the generation process.

4.1.2 *Extending the Set of Terms*

One of the issues that may be encountered while generating a LessLex vector consists in finding just one lexicalization for a given concept ($\mathcal{T}^+$ contains only one element). In such case, the vector for the considered sense would coincide with that of the term in $\mathcal{T}^+$, thus conflating the sense vector and its terminological version from CNN. In order to tackle this issue we try to extend the set of extracted terms by parsing further words from the concept English gloss and WT. In other words, enriching $\mathcal{T}^+$ with further terms is necessary to reshape vectors that have only one associated term as lexicalization. For instance, starting from the term *sunset*^{eng} we encounter the sense `bn:08410678n` (representing the city of Sunset, Texas). This sense is provided with the following lexicalizations:

$$\mathcal{T}^{eng} = \{sunset^{eng}\}; \quad \mathcal{T}^{spa} = \{sunset^{spa}\}; \quad \mathcal{T}^{fra} = \{sunset^{fra}\}.$$

However, out of these three terms only *sunset*^{eng} actually appears in CNN, giving us a final singleton $\mathcal{T}^+ = \{sunset^{eng}\}$. At this point no average can be performed, and the final vector in LessLex for this concept would be identical to the vector of *sunset*^{eng} in CNN. Instead, if we take into consideration the gloss ‘’*Township in Starr County, Texas*‘’, we can extract *township*^{eng} and append it in $\mathcal{T}^+$, thus obtaining a richer vector for this specific sense of *sunset*. In the following sections we describe the two strategies that we developed in order to extract terms from WTs and glosses. The extension strategies are applied for every concept, but in any case, if the final $\mathcal{T}^+$ contains a single term

($|\mathcal{T}^{+}| = 1$), then we discard the sense and we do not include its vector in LessLex.

4.1.2.1 *Extension via Wikipedia Page Title*

The extension via WT only applies to nouns, since senses for other POS are not present in Wikipedia. In detail, if the concept has a Wikipedia Page attached and if the WT provides a disambiguation or specification (e.g., *Chips (company)* or *Magma, Arizona*) we extract the relevant component (by exploiting commas and parentheses of the Wikipedia naming convention) and search for it in CNN. If the whole string cannot be found, we repeat this process by removing the leftmost word of the string until we find a match. In so doing, we search for the maximal sub-string of the WT that has a description in CNN. This allows us to obtain the most specific and yet defined term in CNN. For instance, for the WT *Bat (guided bomb)* we may not have a match in CNN for *guided bomb*, but we can at least add *bomb* to the set of terms in $\mathcal{T}^{+}$.

4.1.2.2 *Extension via Gloss*

Glosses often contain precious pieces of information that can be helpful in the augmentation of the terms associated to a concept. We parse the gloss and extract its components. By construction, descriptions provided in BabelNet glosses can originate from either WordNet or Wikipedia (Navigli and Ponzetto, 2012). In the first case we have (often elliptical) sentences, such as (`bn:00028247n` – *door*) “a swinging or sliding barrier that will close the entrance to a room or building or vehicle”. On the other side, Wikipedia typically provides a plain description like “A door is a panel that makes an opening in a building,

Table 4.1: List of the extraction rules in regex style, describing some POS patterns. If a gloss or a portion of a gloss matches the left part of the rule, then the elements in the right part are extracted. Extracted elements are underlined.

Nouns		
1. to be $\underline{\text{NN+}}$	⟶	$\underline{\text{NN+}}$
2. $\underline{\text{NN1}}$ CC $\underline{\text{NN2}}$	⟶	$\underline{\text{NN1}},\underline{\text{NN2}}$
3. DT $*$ $\underline{\text{NN+}}$	⟶	$\underline{\text{NN+}}$
Verbs		
1. to be $\underline{\text{VB}}$	⟶	$\underline{\text{VB}}$
2. Sentence starts with a $\underline{\text{VB}}$	⟶	$\underline{\text{VB}}$
3. $\underline{\text{VB1}}$ $((\text{CC} \mid ,)$ $\underline{\text{VB2}})+$	⟶	$\underline{\text{VB1}}$, $\underline{\text{VB2+}}$
Adjectives		
1. Sentence is exactly $\underline{\text{JJ}}$	⟶	$\underline{\text{JJ}}$
2. *not* $\underline{\text{JJ}}$	⟶	($\underline{\text{JJ}}$ is dropped)
3. $(\textit{relate} \mid \textit{relating} \mid \textit{related})$ *to* $*$ $\underline{\text{NN}}$	⟶	$\underline{\text{NN}}$
4. $\underline{\text{JJ1}}$ CC $\underline{\text{JJ2}}$	⟶	$\underline{\text{JJ1}},\underline{\text{JJ2}}$
5. $\underline{\text{JJ1}}$, $\underline{\text{JJ2}}$ *or* $\underline{\text{JJ3}}$	⟶	$\underline{\text{JJ1}}$, $\underline{\text{JJ2}}$, $\underline{\text{JJ3}}$

room or vehicle”. Thanks to the regularity of these languages, with few regular expressions on POS patterns[2] we are able to collect enough information to enrich $\mathcal{T}^+$. We devised several rules according to each sense POS; the complete list is reported in Table 4.1 and some applied examples can be found in Table 4.2. In Figure 4.2 we provide an example of the generation process for three concepts, provided by the seed terms $gate^{eng}$ and $gate^{ita}$. For the sake of simplicity, we only show the details regarding two languages (English and Italian). Step *(1)* shows the input terms. In step *(2)* we retrieve three meanings for $gate^{eng}$ and one for $gate^{ita}$, which has already been fetched since it is also a meaning for $gate^{eng}$. For each concept we collect the set of lexicalizations in all considered languages, plus the extensions extracted from WT and gloss.

[2] We adopted the Penn Treebank POS set: https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html.

Table 4.2: Extraction rules applied to different glosses. For each example the sense involved (on the left) and the applied rule (on the right) are reported.

`bn:00012741n` - *Branch*	Noun #2	*A* ***stream*** *or* ***river*** *connected to a larger one.*
`bn:00079944n` - *Winner*	Noun #3	*The* ***contestant*** *who wins the contest.*
`bn:01276497n` - *Plane (river)*	Noun #1	*The Plane is a* ***river*** *in Brandenburg, ...*
`bn:00094850v` - *Tee*	Verb #2	***Connect*** *with a tee.*
`bn:00084198v` - *Build*	Verb #3	***Make*** *by* ***combining*** *materials and parts.*
`bn:00106822a` - *Modern*	Adjective #3	*Relating to a recently developed* ***fashion*** *or style.*
`bn:00103672a` - *Good*	Adjective #4	*Having* ***desirable*** *or* ***positive*** *qualities especially ...*

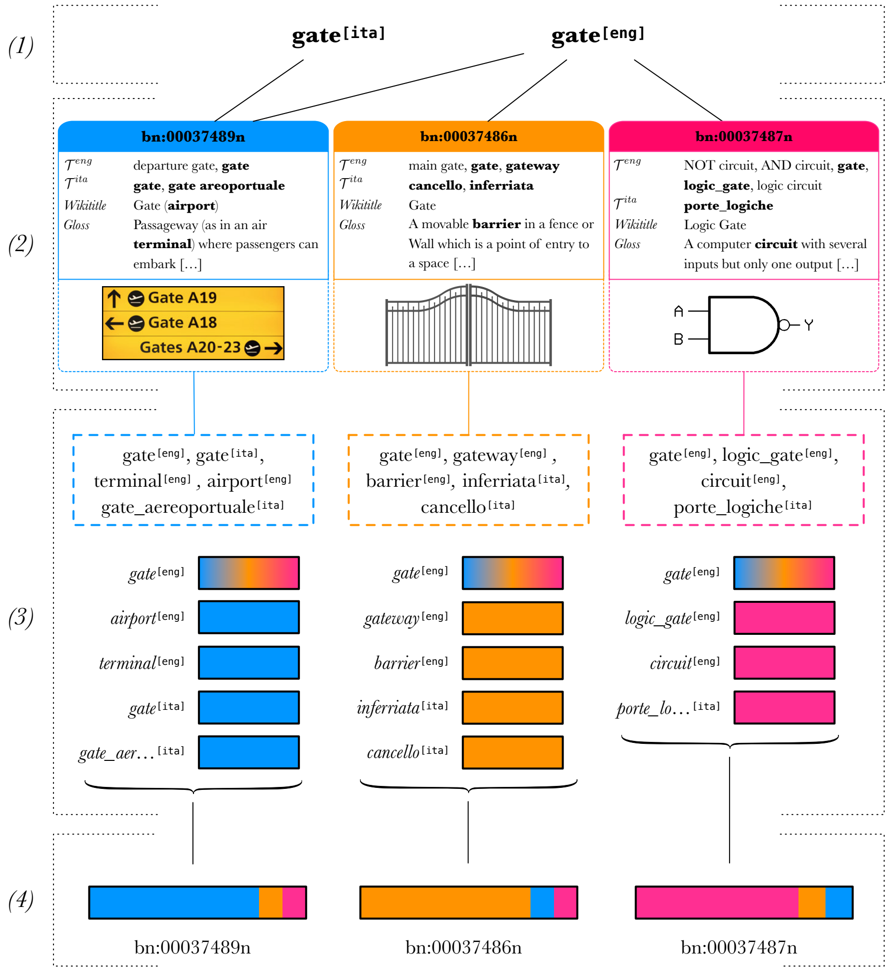

(1)
gate[ita]
gate[eng]
(2)
bn:00037489n
$\mathcal{T}^{eng}$ departure gate, gate
$\mathcal{T}^{ita}$ gate, gate areoportuale
Wikititle Gate (airport)
Gloss Passageway (as in an air terminal) where passengers can embark [...]
Gate A19
Gate A18
Gates A20-23
bn:00037486n
$\mathcal{T}^{eng}$ main gate, gate, gateway
$\mathcal{T}^{ita}$ cancello, inferriata
Wikititle Gate
Gloss A movable barrier in a fence or Wall which is a point of entry to a space [...]
bn:00037487n
$\mathcal{T}^{eng}$ NOT circuit, AND circuit, gate, logic_gate, logic circuit
$\mathcal{T}^{ita}$ porte_logiche
Wikititle Logic Gate
Gloss A computer circuit with several inputs but only one output [...]
A
B
Y
(3)
gate[eng], gate[ita], terminal[eng], airport[eng] gate_aereoportuale[ita]
gate[eng], gateway[eng], barrier[eng], inferriata[ita], cancello[ita]
gate[eng], logic_gate[eng], circuit[eng], porte_logiche[ita]
gate[eng]
airport[eng]
terminal[eng]
gate[ita]
gate_aer…[ita]
gate[eng]
gateway[eng]
barrier[eng]
inferriata[ita]
cancello[ita]
gate[eng]
logic_gate[eng]
circuit[eng]
porte_lo…[ita]
(4)
bn:00037489n
bn:00037486n
bn:00037487n

Figure 4.2: Generation of three LessLex vectors, starting from the seed terms $gate^{eng}$ and $gate^{ita}$.

We then merge all such terms in $\mathcal{T}^+$, by retaining only those that can be actually found in CNN. Once the $\mathcal{T}^+$ sets are computed, we access CNN to retrieve the required vectors for each set *(3)* and then we average them, finally obtaining the vectors for the concepts at hand *(4)*.

We now describe the main features of LessLex, together with the algorithm to compute conceptual similarity on this resource.

4.1.3 LessLex *Features*

The final space in which LessLex vectors reside is an extension of the CNN multilingual semantic space, thus every CNN vector co-exists with the all of the vectors representing its underlying meanings. This peculiar feature allows us to compute the distance between a term and each of its corresponding senses, and such distance can be exploited to determine, given a pair of terms, in which sense they are intended. In Section 4.1.3.2 we will make use of this capability to tackle the word similarity task.

4.1.3.1 LessLex *Statistics*

The LessLex resource[3] has been generated from a group of seed terms collected by starting from $56,322$ words taken from the Corpus of Contemporary American English (COCA) (Davies, 2009) $19,789$ terms fetched from the relevant dictionaries of the Internet Dictionary Project[4] and the $12,544$ terms that appear in the datasets that we used during the evaluation. All terms were POS tagged and duplicates removed beforehand. The final figures of the resource and details concerning its generation are reported in Table 4.3.

We started from a total of $84,620$ terms, and for $65,629$ of them we were able to retrieve at least one sense in BabelNet. The $\mathcal{T}^+$ cardinality shows that our vectors were built by averaging about 6 CNN vectors for each concept. Interestingly, verbs seem to have much richer lexical sets. The final number of senses in LessLex amounts to $174,300$, with a vast majority of nouns. We can also see an interesting overlap between the

[3] The download link for LessLex can be found in Appendix B.
[4] http://www.june29.com/idp/IDPfiles.html.

Table 4.3: Figures on the generation process of LessLex, divided by Part of Speech

LessLex Statistics	All	Nouns	Verbs	Adjectives
Seed terms	84,620	45,297	11,943	27,380
Terms in BabelNet	65,629	41,817	8,457	15,355
$\mathcal{T}^+$ avg. cardinality	6.40	6.16	9.67	6.37
Discarded Senses	16,666	14,737	368	1,561
Unique Senses	174,300	148,380	11,038	14,882
Avg. senses per term	4.80	6.12	3.77	1.77
Total extracted terms	227,850	206,603	8,671	12,576
Avg. extracted terms	1.40	1.46	1.06	1.05

group of senses associated to each term. If we take nouns as example, we have around $42K$ terms providing $148K$ unique senses (3.5 per term), while the average polysemy per term counting repetitions amounts to 6.12. So, we can observe that approximately three senses per term are shared with some other term. A huge amount of concepts are discarded since they only have one term inside $\mathcal{T}^+$: these are named entities or concepts with poor lexicalization sets. The extraction process provided a gran total of about $228K$ terms, and on average each $\mathcal{T}^+$ contains 1.40 additional terms extracted from Wikipedia Page Titles and glosses.

Out of the $117K$ senses in WordNet (version 3.0), roughly $61K$ of them are covered in LessLex. It is however important to note that additional LessLex vectors can be built upon any set of concepts, provided that they are represented in BabelNet (which contains around $15M$ senses) and that some of their lexicalizations are covered in CNN ($1.5M$ terms for the considered languages).

4.1.3.2 *Computing word similarity via ranked-similarity*

As introduced in Section 2.3, the word/concept similarity task consists in determining a score of similarity between two words/concepts provided as input. It was also stated that depending on the type of input (be it conceptual or terminological) the correspondent types of resources are advantaged in the resolution of the task. Finally, when there is no match between the resource and the dataset, the *max-similarity* approach can be exploited. However, since LessLex puts together both terminological and conceptual vectors in one shared space, we decided to develop a novel similarity measure called *ranked-similarity* in order to compute word similarity. Specifically, since we are able to determine not only the distance between each two senses of the input terms, but also the distance between each input term and all of its senses, we use this information to fine tune the computed similarity scores and use ranking as a criterion to grade senses relevance. In particular, we hypothesise that the relevance of senses for a given term can be helpful for the computation of similarity scores, so the ranked-similarity also accounts for the *ranking* of distances between senses and seed term. It implements a heuristics aimed at considering two main elements: the relevance of senses (senses closer to the seed term are preferred), and similarity between sense pairs. Namely, the similarity between two terms t_1, t_2 can be computed as:

$$\text{rnk-sim}(t_1, t_2) = \max_{\substack{\vec{c}_i \in s(t_1)\\ \vec{c}_j \in s(t_2)}} \left[\left((1-\alpha) \cdot (\text{rank}(\vec{c}_i) + \text{rank}(\vec{c}_j))^{-1} \right) + \left(\alpha \cdot \text{cos-sim}(\vec{c}_i, \vec{c}_j) \right) \right], \tag{4.1}$$

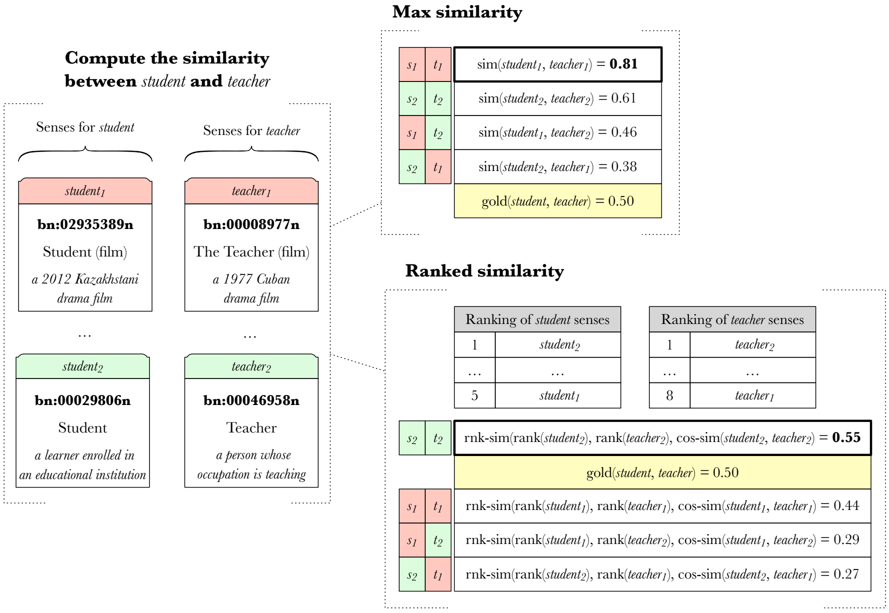

Compute the similarity between *student* and *teacher*
Senses for *student*
Senses for *teacher*
student₁
bn:02935389n
Student (film)
a 2012 Kazakhstani drama film
teacher₁
bn:00008977n
The Teacher (film)
a 1977 Cuban drama film
...
student₂
bn:00029806n
Student
a learner enrolled in an educational institution
teacher₂
bn:00046958n
Teacher
a person whose occupation is teaching
Max similarity
s1 t1 sim(student1, teacher1) = 0.81
s2 t2 sim(student2, teacher2) = 0.61
s1 t2 sim(student1, teacher2) = 0.46
s2 t1 sim(student2, teacher1) = 0.38
gold(student, teacher) = 0.50
Ranked similarity
Ranking of *student* senses
1 student2
5 student1
Ranking of *teacher* senses
1 teacher2
8 teacher1
s2 t2 rnk-sim(rank(student2), rank(teacher2), cos-sim(student2, teacher2) = 0.55
gold(student, teacher) = 0.50
s1 t1 rnk-sim(rank(student1), rank(teacher1), cos-sim(student1, teacher1) = 0.44
s1 t2 rnk-sim(rank(student1), rank(teacher2), cos-sim(student1, teacher2) = 0.29
s2 t1 rnk-sim(rank(student2), rank(teacher1), cos-sim(student2, teacher1) = 0.27

Figure 4.3: A comparison between the max-similarity (Equation 2.2) and the ranked-similarity (Equation 4.1) approaches for the computation of the conceptual similarity.

where α is used to tune the balance between ranking factor and raw cosine similarity.[5] We illustrate the advantages of the ranked similarity with the following example (Figure 4.3). Let us consider the two terms *teacher* and *student*, whose gold-standard similarity score is 0.50.[6] One of the senses of teacher is `bn:02193088n` (*The Teacher (1977 film)* - a 1977 Cuban drama film) while one of the senses of student is `bn:02935389n` (*Student (film)* - a 2012 Kazakhstani drama film). These two senses have a cosine similarity in LessLex of 0.81: such a high score is reasonable, since they are both drama movies. However, it is clear that an annotator would not refer to these two senses for the input terms, but rather to `bn:00046958n` (*teacher* - a person whose occupation is teaching) and `bn:00029806n` (*student* - a learner who is enrolled in an educational institution). These two senses obtain a similarity score of 0.61, which

[5] Presently $\alpha = 0.5$.
[6] We borrow this word pair from the SemEval 17 Task 2 dataset (Camacho-Collados, Pilehvar, Collier, et al., 2017b).

will not be selected since it is lower than 0.81 (as computed through the formula in Equation 2.2). However, if we take into consideration the similarities between the terms *teacher* and *student* and their associated senses, we see that the senses that one would select —while requested to provide a similarity score for the pair— are much closer to the seed terms. The proposed measure involves re-ranking the senses based on their proximity to the term representation, thereby emphasising more relevant terms. We finally obtain similarity of 0.44 for the movie-related senses, while the school-related senses pair obtains a similarity of 0.55, which will be selected and better correlates with human rating.

4.2 EVALUATING LESSLEX

LessLex has been mainly evaluated on the word/concept similarity tasks and then furtherly tested on the contextual similarity task and the semantic text similarity task. We now focus on the first portion of the evaluation by describing the experimental setup and then providing our results and their discussion.

4.2.1 *Experimental Setup*

In this section we introduce the adopted datasets, strategies and other systems employed for the word/concept similarity task.

4.2.1.1 *Adopted Datasets*

In order to properly test the quality of LessLex vectors we considered both conceptual and terminological datasets, ranging on various parts

Table 4.4: List of the dataset employed in the experimentation, showing the POS involved and the languages available in both monolingual and cross-lingual versions.

Dataset	Part of Speech	Monolingual	Cross-lingual
RG-65[1]	nouns	eng, fas, spa	eng, spa, fas, por, fra, deu
WS-Sim-353[2]	nouns	eng, ita, deu, rus	-
SimLex-999[3]	nouns, verbs adjectives	eng, ita, deu, rus	-
SimVerbs-3500[4]	verbs	eng	-
SemEval 17[5]	nouns	eng, deu, ita, spa, fas	eng, deu, ita, spa, fas
Goikoetxea [6]	nouns, verbs adjectives	eus	eng, eus spa, ita

[1] http://lcl.uniroma1.it/similarity-datasets/, https://www.seas.upenn.edu/~hansens/conceptSim/.
[2] http://www.leviants.com/ira.leviant/MultilingualVSMdata.html.
[3] https://fh295.github.io/simlex.html, http://www.leviants.com/ira.leviant/MultilingualVSMdata.html.
[4] http://people.ds.cam.ac.uk/dsg40/simverb.html.
[5] http://alt.qcri.org/semeval2017/task2/index.php?id=data-and-tools.
[6] http://ixa2.si.ehu.es/ukb/bilingual_embeddings.html.

of speech and languages. All benchmarks employed in the experiments are illustrated in Table 4.4.

A pioneering dataset is WordSim-353 (Finkelstein et al., 2002); it has been built by starting from two older sets of word pairs, the RG-65 and MC-30 datasets (Miller and Charles, 1991; Rubenstein and Goodenough, 1965). These dataset were originally conceived for the English language and compiled by human experts. They have then been translated to multilingual and to cross-lingual datasets: the RG-65 has been translated into Farsi and Spanish by Camacho-Collados, Pilehvar, and Navigli (2015a), while the WordSim-353 has been translated by Leviant and Reichart (2015b) into Italian, German and Russian through crowdworkers fluent in such languages. The RG-65 dataset has also been sense-annotated by two humans (Hansen Andrew Schwartz and Gomez, 2011). Additionally, WordSim-353 has been partitioned by individuating the subset of word pairs appropriate for experimenting on similarity judgements rather than on relatedness judgements (Agirre et al., 2009). The SimLex-999 dataset has been compiled through crowd-sourcing, and includes English word pairs covering different parts of speech, namely nouns (666 pairs), verbs (222 pairs) and adjectives (111 pairs) (Hill, Reichart, and Korhonen, 2015). It has been then translated into German, Italian and Russian by Leviant and Reichart (2015a). A dataset has been proposed entirely concerned with English verbs, the SimVerbs-3500 dataset (Gerz et al., 2016); similar to SimLex-999, items herein have been obtained from the USF free-association database (Nelson, McEvoy, and Schreiber, 2004). The SemEval-17 dataset has been developed by Camacho-Collados, Pilehvar, Collier, et al. (2017b); it contains many uncommon entities, like *Si-o-seh pol* or *Mathematical Bridge* encompassing both multilingual and cross-lingual data. Finally, another dataset has been recently released by Goikoetxea, Soroa, and Agirre (2018), in the following referred to as Goikoetxea dataset, built by adding further cross-lingual versions for the RG-65, WS-WordSim-353 and SimLex-999 datasets.

In our evaluation both multilingual and cross-lingual translations have been used. A *multilingual* dataset is one (like RG) where term pairs $\langle x, y \rangle$ from language i have been translated as $\langle x', y' \rangle$ into a different language, such that both x' and y' belong to the same language. An example is $\langle$*casa, chiesa*$\rangle$, $\langle$*house, church*$\rangle$, or $\langle$*maison, église*$\rangle$. Conversely, in a cross-lingual setting (like SemEval 2017, Task 2 - cross-lingual subtask), x' is a term from a language different from that of y', like in the pair $\langle$*casa, church*$\rangle$.

DATASETS ISSUES. Many issues can afflict any dataset, as it is largely acknowledged in literature (Camacho-Collados, Pilehvar, Collier, et al., 2017b; Camacho-Collados, Pilehvar, and Navigli, 2015a; Hill, Reichart, and Korhonen, 2015; E. H. Huang et al., 2012). The oldest datasets are too small (in the order of few tens of word pairs) to attain full statistic significance; until recent years, typically similarity and relatedness (association) judgements have been conflated, thereby penalising models concerned with similarity. Additionally, for such datasets the correlation between systems' results and human rating is higher than human inter-rater agreement. Since human ratings are largely acknowledged as the upper bound to artificial performance in this kind of task, it has been raised that such datasets are not fully reliable benchmarks to investigate the correlation between human judgement and systems' output. Furthermore, a tradeoff exists between the size of the dataset and the quality of the annotation: resources acquired through human experts annotation typically are more limited in size, but featured by higher inter-rater agreement (in the order of .80), while larger datasets suffer from a lower (often with $< .7$) agreement among annotators, thus implying overall reduced reliability. We thus decided to test on all main datasets adopted in literature, to provide the most comprehensive evaluation, widening the experimental base as much as possible. The most recent datasets are in principle more controlled and reliable —SimLex-999, SimVerbs, SemEval-2017, Goikoetxea—, but still we decided to experiment on all of them, since even RG-65 and WS-Sim 353 have been widely used until recently.

4.2.1.2 *Competitors and Resolution Strategies*

As previously stated, different typologies of resources require different strategies in order to compute similarity scores. When dealing with conceptual datasets only the systems that provide conceptual representations are taken in consideration, while the word similarity task is solved by adopting one of four different strategies:

- Max similarity (Equation 2.2): adopted for resources indexed on concepts.
- Cosine similarity: adopted for resources indexed on words.
- Ranked-similarity (Equation 4.1): adopted for resources that share both terminological and conceptual representations in the distributional same space.
- Mf-sense similarity (Most Frequent Sense): adopted as baseline for LessLex in order to better evaluate the effectiveness of the ranked-similarity approach. We select the most frequent sense of the input terms based on the connectivity of the considered sense in BabelNet.The underlying rationale is, in this case, to study how this strategy to pick up senses compares with LessLex vectors, that are built from word embeddings that usually tend to encode the most frequent sense of each word.

Moreover, since the ranked-similarity can be applied only if both input terms are available in CNN (so that we can compute the ranks among their senses), we propose another setup for the usage of LessLex. In the first setup we only make use of the ranked-similarity, so in this setting if at least one given term is not present in CNN we discard the pair

Table 4.5: List of the resources considered in the experimentation and the algorithm we employed for the resolution of the word similarity task.

	Description	Algorithm
LL-M	LessLex	mf-sense similarity
LL-O	LessLex (strategy for handling OOV terms)	ranked-similarity
LLX	LessLex	ranked-similarity
CNN [1]	ConceptNet Numberbatch word embeddings	cosine similarity
NAS [2]	NASARI sense embeddings	max similarity
JCH [3]	JOINTChyb bilingual word embeddings	cosine similarity
SSE [4]	SenseEmbed sense embeddings	max similarity
N2V [5]	NASARI sense embeddings + Word2Vec word embeddings	ranked-similarity

[1] Robyn Speer, Chin, and Havasi (2017) (http://github.com/commonsense/conceptnet-numberbatch v. 16.09)
[2] Camacho-Collados, Pilehvar, and Navigli (2016) (http://lcl.uniroma1.it/nasari/ v. 3.0)
[3] Goikoetxea, Soroa, and Agirre (2018) (http://ixa2.si.ehu.es/ukb/bilingual_embeddings.html)
[4] Iacobacci, Pilehvar, and Navigli (2015) (http://lcl.uniroma1.it/sensembed/)
[5] Word2Vec embeddings trained on UMBC (http://lcl.uniroma1.it/nasari/)

as not covered by the resource. In the second setup (LessLex-OOV, designed to deal with *Out Of Vocabulary* terms) we implemented a fallback strategy to ensure higher coverage: in this case, in order to cope with missing vectors in CNN, we adopt the max-similarity as similarity measure in place of the ranked-similarity.

A summary of the selected competitors and their strategies is reported in Table 4.5. The results obtained by employing LessLex and LessLex-OOV are compared to those obtained by employing NASARI and CNN, to elaborate on similarities and differences with such resources. Additionally, we report the correlation indices obtained by experimenting with other word and sense embeddings that either are trained to perform on specific datasets (JOINTChyb by Goikoetxea, Soroa, and Agirre (2018)), or that directly compare to our resource, as containing both term-level and sense-level vector descriptions (SenseEmbed and NASARI2Vec). A clarification must be done about SenseEmbed. Since in this resource both terminological and sense vectors co-exist in the same space, the application of the ranked-similarity would be fitting. However, in SenseEmbed every sense representation is actually indexed on a pair $\langle term, sense\rangle$, so that different vectors may correspond to a given *sense*. In the ranked-similarity, when computing the distance between a term t and its senses, we retrieve the sense identifiers from BabelNet, so to obtain from SenseEmbed the corresponding vector representations. Unfortunately, however, most senses s_i returned by BabelNet have no corresponding vector in SenseEmbed associated to the term t (i.e., indexed as $\langle t, s_i\rangle$). This fact directly implies a reduced coverage, undermining the performances of SenseEmbed. We then realized that the ranked-similarity is an unfair and not convenient strategy to test on SenseEmbed (in that it forces to use it to some extent improperly), so we resorted to using the max similarity instead.

4.2.2 *Results*

All tables report Pearson and Spearman correlations (denoted by r and ρ, respectively); dashes indicate that a given resource does not deal with the considered input, either because lacking of sense representation, or because lacking of cross-lingual vectors. Similarity values for uncovered pairs were set to the middle point of the similarity scale. Additionally, in Appendix A we report the results obtained by considering only the word pairs covered by all the resources: such figures are of interest, since they allow examining the results obtained from each resource 'in purity', by focusing only on their representational precision.

All top scores are marked with bold fonts.

MULTILINGUAL/CROSS-LINGUAL RG-65 DATASET. The results obtained over the multilingual and cross-lingual RG-65 dataset

Table 4.6: Results on the multilingual and cross-lingual RG-65 dataset, consisting of 65 word pairs. As regards as monolingual correlation scores for the English language, we report results for similarity computed by starting from terms (at *words* level), as well as results with sense identifiers (marked as *senses*). The rest of the results were obtained by using word pairs as input. Reported figures express Pearson (r) and Spearman (ρ) correlations.

RG-65	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
[Word] eng	.64	.59	**.91**	.86	**.91**	.86	**.91**	**.90**	.67	.67	.84	.86	.75	.81	.80	.75
[Sense] eng	-	-	**.94**	**.91**	**.94**	**.91**	-	-	.81	.76	-	-	.72	.76	.78	.73
fas (N)	.75	.72	.75	.75	.73	.70	**.76**	**.76**	.58	.50	-	-	.66	.66	-	-
spa (N)	.82	.82	**.93**	**.93**	**.93**	**.93**	.92	**.93**	.88	.87	.80	.84	.82	.85	-	-
por-fas (N)	.71	.69	.85	.85	.81	.79	**.87**	**.86**	.52	.62	-	-	.70	.66	-	-
fra-por (N)	.82	.83	.92	**.89**	.92	**.89**	**.93**	.88	.69	.67	-	-	.81	.74	-	-
fra-fas (N)	.73	.72	.84	.84	**.86**	.84	**.86**	**.85**	.47	.58	-	-	.72	.71	-	-
fra-spa (N)	.81	.80	**.93**	**.91**	**.93**	**.91**	**.93**	.89	.79	.82	-	-	.88	.86	-	-
fra-deu (N)	.81	.84	**.90**	**.89**	**.90**	**.89**	.88	.87	.77	.77	-	-	.77	.75	-	-
spa-por (N)	.83	.83	**.93**	**.91**	**.93**	**.91**	**.93**	**.91**	.75	.79	-	-	.79	.79	-	-
spa-fas (N)	.71	.70	**.86**	**.87**	.82	.80	**.86**	.86	.50	.64	-	-	.72	.79	-	-
eng-por (N)	.74	.71	**.94**	**.90**	**.94**	**.90**	.92	**.90**	.78	.77	-	-	.80	.76	-	-
eng-fas (N)	.67	.62	**.86**	.85	.84	.81	**.86**	**.87**	.47	.56	-	-	.73	.71	-	-
eng-fra (N)	.71	.70	**.94**	**.92**	**.94**	**.92**	.92	.91	.76	.73	-	-	.81	.75	-	-
eng-spa (N)	.72	.71	**.93**	**.93**	**.93**	**.93**	**.93**	.92	.85	.85	.83	.86	.80	.85	-	-
eng-deu (N)	.74	.72	**.91**	**.89**	**.91**	**.89**	.89	**.89**	.70	.74	-	-	.76	.80	-	-
deu-por (N)	.87	.84	**.91**	**.87**	**.91**	**.87**	**.91**	**.87**	.73	.76	-	-	.76	.72	-	-
deu-fas (N)	.77	.74	.85	**.85**	**.87**	.84	.85	.84	.58	.65	-	-	.78	.80	-	-
deu-spa (N)	.84	.85	**.91**	**.90**	**.91**	**.90**	.90	.89	.71	.79	-	-	.79	.80	-	-

are illustrated in Table 4.6. RG-65 includes a multilingual dataset and a cross-lingual one. As regards as the former one, both LessLex and LessLex-OOV obtain analogous correlation with respect to CNN when considering term pairs; LessLex and LessLex-OOV substantially outperform NASARI, SenseEmbed and NASARI2Vec while considering sense pairs (Hansen Andrew Schwartz and Gomez, 2011). Of course CNN is not evaluated in this setting, since it only includes representations for terms. As regards as the latter subset, containing cross-lingual files, figures show that both CNN and LessLex obtained high correlations, higher than the competing resources providing meaning representations for the considered language pairs.

Table 4.7: Results on the WS-Sim-353 dataset, where we experimented on the 201 word pairs (out of the overall 353 elements) that are acknowledged as appropriated for computing similarity. Reported figures express Pearson (r) and Spearman (ρ) correlations.

WS-Sim-353	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N)	.67	.65	**.78**	.78	**.78**	.78	**.78**	**.79**	.60	.61	.72	.72	.69	.73	.71	.70
ita (N)	.67	.68	.70	.73	**.74**	**.78**	.69	.73	.66	.65	.60	.62	.66	.73	-	-
deu (N)	.73	.71	.63	.68	.76	.77	**.82**	**.81**	.64	.63	-	-	.62	.60	-	-
rus (N)	.72	.70	.64	.62	**.73**	**.75**	.65	.63	.63	.61	-	-	.60	.60	-	-

MULTILINGUAL WS-SIM-353 DATASET. The results on the multilingual WS-Sim-353 dataset are presented in Table 4.7. Results on this data differ according to the considered language: interestingly enough, for the English language, the results computed via LessLex are substantially on par with those obtained by employing CNN vectors. As regards as the remaining translations of the dataset, CNN and LessLex achieve the highest correlations also on the Italian, German and Russian languages. Different from other experimental settings (see, e.g., the RG-65 dataset), the differences in correlation are more consistent, with LessLex obtaining top correlation scores for Italian and Russian, and CNN for German.

MULTILINGUAL SIMLEX-999 DATASET. The results obtained on the SimLex-999 dataset are reported in Table 4.8. We face here twofold results: as regards as the English and the Italian translation, we recorded better results when using the LessLex vectors, with consistent advantage over competitors on English verbs. As regards as English adjectives, the highest correlation was recorded when employing the LessLex Most Frequent Sense vectors (LL-M column). As regards as Italian, as in the WordSim-353 dataset, the LessLex-OOV strategy obtains correlations with human ratings that are higher or on par with

Table 4.8: Results on the multilingual SimLex-999, including overall 999 word pairs, with 666 nouns, 222 verbs and 111 adjectives for the English, Italian, German and Russian languages. Reported figures express Pearson (r) and Spearman (ρ) correlations.

SimLex-999	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N)	.51	.50	**.69**	**.67**	**.69**	**.67**	.66	.63	.40	.38	.55	.53	.52	.49	.46	.43
eng (V)	.62	.56	**.67**	**.65**	**.67**	**.65**	.61	.58	-	-	.51	.50	.54	.49	-	-
eng (A)	**.84**	**.83**	.82	.79	.82	.79	.80	.78	-	-	.63	.62	.55	.51	-	-
eng (*)	.57	.55	**.70**	**.69**	**.70**	**.69**	.67	.65	-	-	.55	.54	.53	.49	-	-
ita (N)	.50	.49	**.66**	**.63**	.64	**.63**	.64	.61	.45	.46	.47	.47	.56	.49	-	-
ita (V)	.58	.52	**.69**	**.63**	**.69**	**.63**	.67	.58	-	-	.54	.47	.54	.44	-	-
ita (A)	.65	.58	**.74**	**.69**	**.74**	**.69**	**.74**	.66	-	-	.39	.30	.57	.47	-	-
ita (*)	.51	.47	**.66**	**.62**	.65	**.62**	.65	.61	-	-	.46	.44	.54	.47	-	-
deu (N)	.58	.56	.65	.63	.65	.64	**.66**	**.65**	.41	.42	-	-	.47	.43	-	-
deu (V)	.48	.42	.54	.45	.54	.46	**.63**	**.57**	-	-	-	-	.43	.37	-	-
deu (A)	.66	.63	.66	.65	.69	.68	**.77**	**.75**	-	-	-	-	.43	.26	-	-
deu (*)	.55	.52	.62	.59	.63	.61	**.67**	**.65**	-	-	-	-	.45	.38	-	-
rus (N)	.43	.42	.52	.48	.51	**.50**	**.53**	.48	.20	.22	-	-	.26	.21	-	-
rus (V)	.31	.19	.25	.18	.27	.20	**.60**	**.55**	-	-	-	-	.23	.20	-	-
rus (A)	.25	.26	.25	.25	.27	.28	**.69**	**.69**	-	-	-	-	.04	.04	-	-
rus (*)	.36	.32	.43	.37	.42	.39	**.56**	**.51**	-	-	-	-	.23	.13	-	-

Table 4.9: Results on the SimVerbs-3500 dataset, containing 3,500 verb pairs. Reported figures express Pearson (r) and Spearman (ρ) correlations.

SimVerbs	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (V)	.58	.56	**.67**	**.66**	**.67**	**.66**	.62	.60	-	-	.56	.56	.45	.42	.31	.30

respect to those obtained by using LessLex vectors. In the second half of the dataset CNN performed better on German and Russian.

SIMVERBS-3500 DATASET. Results obtained while testing on the SimVerbs-3500 dataset are reported in Table 4.9. In this case it is straightforward to notice that the results obtained by LessLex outperform those by all competitors, with a gain of .05 in Pearson r, and .06 in Spearman correlation over CNN, on this large set of 3500 verb pairs. It was not possible to use NASARI vectors, that only exist for

Table 4.10: Results on the SemEval 17 Task 2 dataset, containing 500 noun pairs. Reported figures express Pearson (r) and Spearman (ρ) correlations.

SemEval 17	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N)	.71	.72	**.79**	.80	.77	**.81**	**.79**	.79	.64	.65	.50	.45	.69	.73	.64	.64
deu (N)	**.73**	.72	.69	.68	.71	**.75**	.70	.68	.62	.62	-	-	.60	.61	-	-
ita (N)	.74	.75	.66	.65	**.76**	**.79**	.63	.61	.72	.73	.54	.50	.70	.73	-	-
spa (N)	**.77**	.79	.67	.66	.74	**.80**	.63	.62	.72	.73	.50	.48	.68	.71	-	-
fas (N)	.67	.67	.43	.47	**.72**	**.75**	.39	.35	.54	.53	-	-	.60	.63	-	-
deu-spa (N)	**.76**	.77	.69	.68	.74	**.79**	.66	.64	.54	.55	-	-	.65	.68	-	-
deu-ita (N)	**.75**	.76	.68	.67	**.75**	**.79**	.65	.63	.53	.65	-	-	.62	.62	-	-
eng-deu (N)	**.75**	.75	**.75**	.75	**.75**	**.79**	.74	.73	.51	.62	-	-	.63	.63	-	-
eng-spa (N)	.75	.76	.73	.73	**.76**	**.82**	.70	.70	.66	.70	.46	.44	.59	.61	-	-
eng-ita (N)	.74	.76	.72	.72	**.76**	**.82**	.69	.69	.63	.71	.38	.36	.69	.73	-	-
spa-ita (N)	**.76**	.77	.67	.66	**.76**	**.81**	.63	.61	.65	.72	.41	.39	.59	.61	-	-
deu-fas (N)	.72	.73	.55	.52	**.73**	**.76**	.51	.47	.39	.52	-	-	.63	.65	-	-
spa-fas (N)	.72	.73	.55	.52	**.75**	**.79**	.50	.47	.47	.61	-	-	.66	.70	-	-
fas-ita (N)	.72	.73	.53	.50	**.75**	**.78**	.49	.45	.43	.58	-	-	.66	.69	-	-
eng-fas (N)	.71	.72	.58	.55	**.74**	**.79**	.54	.51	.42	.59	-	-	.67	.70	-	-

noun senses; also notably, the results obtained by employing the baseline (LL-M) strategy outperformed those obtained through SenseEmbed and NASARI2Vec.

SEM EVAL 17 TASK 2 DATASET. The figures obtained by experimenting on the "SemEval 17 Task 2: Multilingual and Cross-lingual Semantic Word Similarity" dataset are provided in Table 4.10. This benchmark is a multilingual dataset including 500 word pairs (nouns only) for monolingual versions, and 888 to 978 word pairs for the cross-lingual ones.

These results are overall favorable to LessLex in the comparison with CNN and with all other competing resources. Interestingly enough, while running the experiments with CNN vectors we observed even higher correlation scores than those obtained in the SemEval 2017 evaluation campaign (Camacho-Collados, Pilehvar, Collier, et al., 2017b; Robyn

Speer, Chin, and Havasi, 2017). At that time, such figures scored highest on all multilingual tasks (with the exception of the Farsi language) and on all cross-lingual settings (with no exception). To date, as regards as the cross-lingual setting, LessLex correlations indices are constantly higher than those by competitors, including CNN. We observe that the scores obtained by employing the baseline with most frequent senses (LL-M) are always ameliorative with respects to all results obtained by experimenting with NASARI, JOINTChyb, SenseEmbed and NASARI2Vec (with the only exception of the ρ score obtained by SSE on the English monolingual dataset).

MULTILINGUAL/CROSSLINGUAL GOIKOETXEA DATASET. The results obtained by testing on the Goikoetxea dataset are reported in Table 4.11. The dataset includes new variants for three popular dataset: three cross-lingual versions for the RG-65 dataset (including the Basque language, marked as 'eus' in the Table); the six cross-lingual combinations of the Basque, Italian and Spanish translations of the WS-Sim-353 dataset; and three cross-lingual translations of the SimLex-999 dataset, including its English, Italian and Spanish translations.

Results are thus threefold. As regards as the first block on the RG-65 dataset, LessLex results outperform all competitors (to a smaller extent on versions involving the Basque language), including JOINTChyb, the best model by Goikoetxea, Soroa, and Agirre (2018). In the comparison with CNN, LessLex vectors achieve better results, with higher correlation for cases involving Basque, on par on the English-Spanish dataset. As regards as the second block (composed of cross-lingual translations of the WS-Sim-353 dataset), we record that the LessLex-OOV strategy

Table 4.11: Results on the Goikoetxea dataset. The dataset includes variants of the RG-65 (first block), WS-Sim-353 (second block) and SimLex-999 (third block) datasets. The 'eus' abbreviation indicates the Basque language. Reported figures express Pearson (r) and Spearman (ρ) correlations.

Goikoetxea	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
spa-eus (N)	.74	.72	.42	.67	**.76**	**.77**	.66	.61	.71	.74	.73	.72	.61	.71	-	-
eng-eus (N)	.74	.74	.41	.77	**.89**	**.91**	.77	.73	**.89**	.88	.88	.87	.81	.83	-	-
eng-spa (N)	.72	.71	**.93**	**.93**	**.93**	**.93**	**.93**	**.93**	.77	.82	.83	.86	.64	.85	-	-
eus-ita (N)	.27	.68	.42	**.74**	.24	.71	.51	.53	.49	.56	**.52**	.58	.20	.58	-	-
spa-ita (N)	.29	.66	.29	**.76**	.29	.74	**.63**	.70	.53	.57	.54	.60	.21	.59	-	-
spa-eus (N)	.31	.74	.40	**.78**	.29	**.78**	.55	.56	.59	.66	**.69**	.73	.23	.64	-	-
eng-ita (N)	.30	.64	.27	**.77**	.32	.76	**.67**	.74	.47	.52	.59	.64	.21	.59	-	-
eng-eus (N)	.30	.70	.39	**.79**	.29	.78	.56	.57	.52	.60	**.71**	.75	.23	.64	-	-
eng-spa (N)	.34	.66	.27	**.79**	.40	.77	**.70**	.76	.52	.56	.68	.73	.29	.64	-	-
eng-spa (N)	.49	.48	**.66**	**.64**	.65	**.64**	.64	.62	.36	.46	.54	.51	.53	.50	-	-
eng-spa (V)	.54	.50	.61	.59	**.62**	**.60**	.58	.56	-	-	.43	.43	.52	.49	-	-
eng-spa (A)	.72	.73	.73	.74	.72	**.75**	**.74**	.74	-	-	.56	.55	.53	.47	-	-
eng-spa (*)	.53	.51	**.66**	.64	.65	**.65**	.64	.63	-	-	.50	.52	.53	.49	-	-
eng-ita (N)	.52	.52	**.70**	**.68**	**.70**	**.68**	.68	.66	.36	.45	.51	.50	.54	.51	-	-
eng-ita (V)	.49	.40	.57	.51	.57	.51	**.67**	**.62**	-	-	.47	.51	.44	.33	-	-
eng-ita (A)	.75	.74	**.79**	**.78**	**.79**	**.78**	.77	.72	-	-	.42	.43	.57	.45	-	-
eng-ita (*)	.50	.46	.65	.62	.65	.63	**.68**	**.66**	-	-	.48	.50	.51	.43	-	-
spa-ita (N)	.53	.53	**.67**	.65	**.67**	**.66**	.66	.64	.34	.45	.45	.45	.54	.52	-	-
spa-ita (V)	.44	.39	.51	.46	.51	.46	**.63**	**.60**	-	-	.42	.44	.43	.34	-	-
spa-ita (A)	.68	.66	**.73**	.71	.72	**.73**	**.73**	.69	-	-	.41	.45	.57	.48	-	-
spa-ita (*)	.49	.46	.61	.58	.61	.59	**.66**	**.64**	-	-	.44	.45	.50	.45	-	-

obtained the top Spearman correlation scores, coupled to poor Pearson correlation scores; while CNN and JCH obtain the best results as regards as the latter coefficients. As regards as the last block of results in Table 4.11 (containing translations for the SimLex-999 dataset), we first observe that comparing the obtained figures is not simple: we report the figures obtained by Goikoetxea, Soroa, and Agirre (2018) with no distinction in POS. However, if we focus on results on nouns (two thirds of the SimLex-999 dataset), LessLex vectors obtain the best results, while it is not easy to determine whether LessLex or CNN vectors provided the overall best results on the other parts of speech.

4.2.2.1 *Discussion*

We overall experimented on nine different languages (deu, eng, eus, fas, fra, ita, por, rus, spa) and various cross-lingual combinations. Collectively, such tests constitute a widely varied experimental setting, to the best of our knowledge the largest on the semantic similarity task. The obtained results authorise to state that LessLex is at least on par with competing state-of-the-art resources, although we also noticed that some room still exists for further improvements, such as the coverage on individual languages (e.g., Russian and German).

Let us start by considering the results on the multilingual WS-Sim-353 and on the SimLex datasets (Tables 4.7 and 4.8, respectively). The results obtained through LessLex always improve on those obtained by employing the sense embeddings by SenseEmbed and NASARI2Vec, that provide term and sense descriptions embedded in the same semantic space, and are thus closer to our resource. Also the comparison with NASARI is favorable to LessLex. In the comparison with CNN, we note that while in the English language LessLex and LessLex-OOV scores either outperform or closely approach those obtained through CNN, in other languages our vectors suffer from the reduced and less rich sense inventory of BabelNet, that in turn determines a lower quality for our vectors. This can be easily figured if one considers that a less rich synset contains less terms to be plugged into our vectors, thereby determining an overall poorer semantic coverage. The poor results obtained by employing LessLex on the German and Russian subsets of the WS-Sim-353 and SimLex-999 datasets probably stem from this sort of limitation.

A consistent difference between LessLex ranked-similarity and the LessLex-OOV strategy can be observed when a sense is available in BabelNet, but not the corresponding vector in CNN: the LessLex-OOV strategy basically consists in resorting to the maximization approach when —due to the lack of a terminological description associated to the sense at hand— it is not possible to compute the ranked-similarity. This strategy was executed in around 9% of cases ($\sigma = 12\%$) over all datasets, ranging from 0% on verbs in the SimVerbs-3500 dataset, up to around 50% for the Farsi nouns in the SemEval-2017 monolingual dataset. Although not employed often, this strategy contributed in many cases to obtain top scoring results, improving on those computed with plain ranked-similarity with LessLex, and also in some cases on CNN and NASARI, as illustrated in both the monolingual and cross-lingual portions of the SemEval-2017 dataset (Table 4.10).

Cases where results obtained through LessLex improve over those obtained with CNN are important to assess LessLex, in that they confirm that the control strategy for building our vectors is effective, and that our vectors contain precise and high quality semantic descriptions. In this sense, obtaining higher or comparable results by using sense embeddings with respect to using word embeddings (with sense embeddings featuring an increased problem space with respect to the latter ones) is *per se* an achievement. Additionally, our vectors are grounded on BabelNet synset identifiers, which allows to address each sense as part of a large semantic network, providing further information on senses with respect to the meaning descriptions conveyed through the 300-dimensional vectors. While the LessLex-OOV is a run-time strategy concerned with the usage of LessLex to compare sense pairs, the quality of our vectors is determined by the enrichment step. More specifically, the coverage of our vectors depends on the strategy devised to build $\mathcal{T}^+$ because the coverage is determined both by the number of term-level vectors, and by the number of sense vectors associated to each term, so that in a sense the coverage of LessLex is determined by the size of $\mathcal{T}^+$. Additionally, we register that the elements added to the extended set $\mathcal{T}^+$ are often of high quality, as proven, for example, by the sense-oriented task of the RG-65 dataset, where senses were assessed (Table 4.6, line 2): in this setting, the correlation indices for LessLex and LessLex-OOV vectors score highest over all semantic resources, including NASARI, SenseEmbed and NASARI2Vec.

Also results achieved while testing on the Goikoetxea dataset seem to confirm that our LL-O strategy allows to deal with languages with reduced (with respect to English) coverage and/or sense inventory in either BabelNet or ConceptNet: in 12 out of the overall 18 tests on this dataset, the LessLex-OOV strategy earned at least one top scoring correlation index (either r or ρ, as shown in Table 4.11). The comparison with the recent JOINTChyb embeddings shows that the adoption of a shared conceptual —multilingual— level can be beneficial and advantageous with respect to building specialised pairs of embeddings.

Less relevant under a cross-lingual perspective, but perhaps relevant in order to fully assess the strengths of our resource, LessLex vectors achieved by far highest correlation scores on English verbs (please refer to Table 4.8, line 2 and Table 4.9). The comparison with previous literature seems to corroborate this fact (Gerz et al., 2016): in fact, to the best of our knowledge previous state-of-the-art systems achieved around .624 Spearman correlation (Faruqui and Dyer, 2015; Mrkšić et al., 2016).

In order to further deepen the analysis of results, it is instructive to compare the results reported in Tables 4.6-4.11 with those obtained on the fraction of dataset covered by all considered resources, and provided in Appendix A (Tables A.1-A.6). That is, for each dataset we re-run the experiments for all considered resources by restricting to compare only term pairs actually covered by all resources. We will call this evaluation metrics *CbA condition* hereafter (from 'Covered by All'); as opposed to the case in which a mid-scale similarity value was assigned to uncovered terms, referred to as *MSV condition* in the following (from 'Mid Scale Value'). As mentioned, the CbA condition allows evaluating the representational precision of the resources at stake independent of their coverage, whilst a mixture of both aspects is grasped in the the MSV condition. In the leftmost column of Tables in Appendix A we report the coverage for each test. As we can see, coverage is diverse across datasets, ranging from .61 (averaged on all variants, with a minimum on the Farsi language, in the order of .34 and all translations involving the Farsi) in the SemEval-2017 dataset (Table A.5) to 1.0 in the SimVerbs-3500 dataset (Table A.3). Other notable cases in which relevant variations in coverage were observed are Russian verbs and adjectives in the SimLex-999 dataset, with .20 and .06 coverage, respectively (Table A.4). In general, as expected, the recorded correlations are improved with respect to results registered for the corresponding (same dataset and resource) test in the MSV setup, although spot pejorative cases were observed, as well (see, e.g., CNN results for Italian adjectives, in the SimLex-999 dataset, reported in

Table 4.12: The top half Table shows a synthesis of the results obtained in the Mid-Scale similarity Value (MSV) experimental condition, whose details have been illustrated in Tables 4.6-4.11; at the bottom we provide a synthesis of the results obtained in the Covered by All (CbA) experimental condition, illustrated in detail in Tables A.1-A.6.

Mid-Scale similarity Value (MSV) Experimental Condition

	LL-M	LLX	LL-O	CNN	NAS	JCH	SSE	N2V
Spearman ρ	7	32	41	33	1	3	0	0
Pearson r	1	32	50	24	0	0	0	0
Total	8	64	91	57	1	3	0	0

Covered by All (CbA) Experimental Condition

	LL-M	LLX	LL-O	CNN	NAS	JCH	SSE	N2V
Spearman ρ	1	61	-	30	0	0	0	0
Pearson r	2	63	-	22	0	0	0	0
Total	3	124	-	52	0	0	0	0

Table A.4). For example, if we consider the poorly covered SemEval-2017 dataset, we observe the following rough improvements (average over all translations, and both r and ρ metrics) in the correlation indices: .20 for LessLex, .22 for CNN, .09 for NASARI, .30 for JOINTChyb (that does not cover all translations, anyway), .07 for SenseEmbed, and .09 for NASARI2Vec (only dealing with nouns).

In order to synthetically examine how the CbA experimental condition affected results with respect to the MSV condition, we adopt a rough index, simply counting the number of test results (we consider as a separate test result each Pearson and each Spearman score in Tables A.1-A.6) where each resource obtained highest scores.[7] We thus count overall 152 tests (15 in the SemEval-2017 dataset, 4 in the WS-Sim-353, 1 in the SimVerbs-3500, 16 in the SimLex-999, 19 in the RG-65, and 21 in the Goikoetxea; for each one we consider as separated r and ρ scores). Provided that in several cases we recorded more than one single resource attaining top scores, the impact of the reduced coverage (CbA condition)

[7] Of course we are aware that this is only a rough index, that e.g., does not account for the datasets size (varying from 65 to 3,500 word pairs) or the involved POS, and mixing Pearson and Spearman correlation scores.

vs. MSV condition is presented in Table 4.12. In the MSV condition we have LessLex-OOV achieving 91 top scoring results, followed by LessLex with 64 and CNN with 57. In the CbA experimental condition, the LessLex-OOV strategy was never executed (since only the actual coverage of all resources was considered, and no strategy for handling out-of-vocabulary terms was thus necessary), and LessLex obtained 124 top scoring results, against 52 for CNN. In the latter condition there were less cases with a tie. All in all, we interpret the different correlation scores obtained in the two experimental conditions as an evidence that LessLex embeddings are featured by good coverage (as suggested by the results obtained in the MSV condition) and lexical precision (as suggested by the results obtained in the CbA condition), improving on those provided by all other resources at stake.

Our approach showed to scale well to all considered languages, under the mild assumption that these are covered by BabelNet, and available in the adopted vectorial resource; when such conditions are met, LessLex vectors can be in principle built on a streamlined, on-demand, basis, for any language and any POS.

4.3 USING LESSLEX

It is acknowledged that the intrinsic evaluation via word similarity can be sometimes not sufficient to assess the quality of embeddings (Chiu, Korhonen, and Pyysalo, 2016). For this reason we decided to implement a deeper evaluation and to study how LessLex performs when employed to tackle two extrinsic tasks, namely the Contextual Similarity and the Semantic Text Similarity.

4.3.1 LessLex *and Contextual Similarity*

The contextual similarity task is a variant of the word similarity task in which the two words in input are given *in context*, meaning that they are presented together with the piece of text in which they occur. We tested on two different datasets, namely the Stanford's Contextual Word Similarities Datastet (SCWS) (E. H. Huang et al., 2012), and on the more recent Word-in-Context Dataset (WiC) (Pilehvar and Camacho-Collados, 2019). In the following we report the results obtained on the two datasets by experimenting with LessLex and NASARI2Vec, which is the only competing resource suitable to implement the ranked similarity along with its contextual variant.

4.3.1.1 *Contextual Similarity on SCWS*

The SCWS dataset defines the problem as a similarity task, where each input record contains two sentences in which two distinct target words t_1 and t_2 are used. The task requires to provide the pair $\langle t_1, t_2 \rangle$ with a similarity score by taking into account the context where the given terms occur. The dataset consists of $2,003$ instances, divided into $1,328$ instances whose targets are a noun pair, 399 a verb pair, 97 adjectival pair, 140 contain a verb-noun pair, 30 contain a noun-adjective pair, and 9 a verb-adjective pair. To test on the SCWS dataset we employed both the ranked-similarity (rnk-sim) and the *contextual* ranked-similarity (c-rnk-sim), a variant specifically devised to account for contextual information. As regards as the latter one, given two sentences $\langle S_1, S_2 \rangle$, we first computed the context vectors $\langle \overrightarrow{ctx}_1, \overrightarrow{ctx}_2 \rangle$ with a bag-of-words

Table 4.13: Results obtained by experimenting on the SCWS dataset. Figures report the *Spearman* correlations with the gold standard divided by part of speech. In the top of table we report our own experimental results, while in the bottom results from literature are provided.

System	ALL	N-N	N-V	N-A	V-V	V-A	A-A
LessLex (rnk-sim)	0.695	0.692	0.696	0.820	0.641	0.736	0.638
LessLex (c-rnk-sim)	0.667	0.665	0.684	0.744	0.643	0.725	0.524
NASARI2Vec (rnk-sim)	-	0.384	-	-	-	-	-
NASARI2Vec (c-rnk-sim)	-	0.471	-	-	-	-	-
SenseEmbed[1]	0.624	-	-	-	-	-	-
Huang et al. 50d[2]	0.657	-	-	-	-	-	-
Arora at al.[3]	0.652	-	-	-	-	-	-
MSSG.300D.6K[4]	0.679	-	-	-	-	-	-
MSSG.300D.30K[4]	0.678	-	-	-	-	-	-

[1] Iacobacci, Pilehvar, and Navigli (2015)
[2] E. H. Huang et al. (2012)
[3] Arora et al. (2018)
[4] Neelakantan et al. (2014), figures reported from Mu, Bhat, and Viswanath (2017)

approach, that is by averaging all the terminological vectors of the lexical items contained therein:

$$\overrightarrow{ctx}_i = \frac{\sum_{t \in S_i} \vec{t}}{N} \tag{4.2}$$

where N is the number of words in the sentence S_i. The two context vectors are then used to perform the sense rankings for the target words, in the same fashion as in the original ranked-similarity:

$$\text{c-rnk-sim}(t_1, t_2, \overrightarrow{ctx}_1, \overrightarrow{ctx}_2) = \max_{\substack{\vec{c}_i \in s(t_1) \\ \vec{c}_j \in s(t_2)}} \left[\left((1-\alpha) \cdot \big(\underbrace{\text{rank}(\vec{c}_i)}_{\text{w.r.t. } \overrightarrow{ctx}_1} + \underbrace{\text{rank}(\vec{c}_j)}_{\text{w.r.t. } \overrightarrow{ctx}_2} \big)^{-1} \right) + \left(\alpha \cdot \text{cos-sim}(\vec{c}_i, \vec{c}_j) \right) \right]. \tag{4.3}$$

results The results obtained are reported in Table 4.13.[8] In spite of the simplicity of the system employing LessLex embeddings,

[8] Parameters setting: in rnk-sim and in the c-rnk-sim α was set to 0.5 for both LessLex and NASARI2Vec.

Table 4.14: Correlation scores obtained with LessLex on different subsets of data obtained by varying standard deviation in human ratings. The reported figures show higher correlation when testing on the most reliable (with smaller standard deviation) portions of the dataset. To interpret the standard deviation values, we recall that the original ratings collected in the SCWS dataset were expressed in the range $[0.0, 10.0]$.

σ	c-rank-sim (r)	rank-sim (r)	nof-items
≤ 0.5	0.83	0.82	39
≤ 1.0	0.85	0.86	82
≤ 1.5	0.85	0.85	165
≤ 2.0	0.82	0.84	285
≤ 2.5	0.68	0.83	518
≤ 3.0	0.68	0.79	903
≤ 3.5	0.67	0.75	$1,429$
≤ 4.0	0.64	0.71	$1,822$
< 5.0	0.63	0.69	$2,003$

our results overcome those reported in literature, where by far more complex architectures were used. However, such scores are higher than the agreement among human raters, which can be thought of as an upper bound to systems' performance. The Spearman correlation among human ratings (computed on leave-one-out basis, that is by averaging the correlations between each rater and the average of all other ones) is reportedly of 0.52 for the SCWS dataset (Chi and Y.-N. Chen, 2018; Chi, Shih, and Y.-N. Chen, 2018), which can be considered as a poor inter-rater agreement. Despite this fact, SCWS is considered one of the standard benchmarks for the task and some interesting insights can be still be drawn from this experimentation. Also to some extent surprising is the fact that the simple ranked-similarity (rnk-sim), which was intended as a plain baseline, surpassed the contextual ranked-similarity (c-rnk-sim), more suited for this task.

To further elaborate on our results we then re-run the experiment by investigating how the obtained correlations are affected by different degrees of consistency in the annotation. We partitioned the dataset items based on the standard deviation recorded in human ratings, obtaining 9 bins, and re-run our system on these, utilizing both metrics, with same parameter settings as in the previous run. In this case the Pearson correlation indices were recorded, in order to investigate the linear relationship between our output and human ratings. As expected, we obtained higher correlations on the most reliable portions of the dataset, those with smallest standard deviation (Table 4.14).

However, we still found surprising the obtained results, since the rnk-sim metrics seems to be more robust than its contextual counterpart. This is in contrast with literature, where the top scoring metrics, originally defined by Reisinger and Mooney (2010), also leverage contextual information (T. Chen et al., 2015; X. Chen, Z. Liu, and Sun, 2014; E. H. Huang et al., 2012). In particular, the *AvgSim* metrics (which is computed as a function of the average similarity of all prototype pairs, without taking into account the context) is reportedly outperformed by the *AvgSimC* metrics, in which terms are weighted by the likelihood of the word contexts appearing in the respective clusters). The *AvgSim* and the *AvgSimC* directly compare to our rnk-sim and c-rnk-sim metrics, respectively. In our results, for the lowest levels of standard deviation (that is, for $\sigma \leq 2$), the two metrics perform in similar way; for growing values of σ we observe a substantial drop of the c-rank-sim, while the correlation of the rnk-sim decreases more smoothly. In these cases (for

Table 4.15: Some descriptive statistics of the WiC dataset. In particular, the distribution of nouns and verbs, number of instances and unique words across training, development and test-set of the WiC dataset are reported.

Split	**Instances**	**Nouns**	**Verbs**	**Unique Words**
Training	5,428	49%	51%	1,256
Dev	638	62%	38%	599
Test	1,400	59%	41%	1,184

$\sigma \geq 2.5$) contextual information seems to be less relevant than pair-wise similarity of term pairs taken in isolation.

4.3.1.2 *Contextual Similarity on WiC*

In the WiC dataset the contextual word similarity problem is cast to a binary classification task: each instance is composed of two sentences in which a specific target word t is used. The employed algorithm has to make a decision on whether t assumes the same meaning or not in the two given sentences. The distribution of nouns and verbs across training, development and test-set is reported in Table 4.15, together with figures on number of instances and unique words.

Different from the SCWS dataset, in experimenting on WiC we are required to decide whether a given term conveys same or different meaning in their context, as in a binary classification task. Context-insensitive word embedding models are expected here to approach a random baseline, while the upper bound, provided by human-level performance, is 80% accuracy.

We run two experiments, one where the contextual ranked-similarity was employed, the other with the Rank-Biased Overlap (Webber, Moffat, and Zobel, 2010). In the former case, we used the *contextual* ranked-similarity (Equation 4.3) as the metrics to compute the similarity score, and we added a similarity threshold to provide a binary answer. In the latter case, we designed another simple schema to assess the semantic similarity between term senses and context. At first we built a context vector (Equation 4.2) to acquire a compact vectorial description of both texts at hand, obtaining two context vectors $\overrightarrow{ctx_1}$ and $\overrightarrow{ctx_2}$. We then ranked all senses of the term of interest (based on the cosine similarity metrics) with respect to both context vectors, obtaining s_1^t and s_2^t, as the similarity ranking of t senses from $\overrightarrow{ctx_1}$ and $\overrightarrow{ctx_2}$, respectively. The Rank-Biased Overlap (RBO) metrics was then used to compare the similarity between such rankings. Given two rankings s_1^t and s_2^t, RBO is defined as follows:

$$\text{RBO}(s_1^t, s_2^t) = (1-p) \sum_{d=1}^{|O|} p^{d-1} \frac{|O_d|}{d}, \tag{4.4}$$

where O is the set of overlapping elements, $|O_d|$ counts the number of overlaps out of the first d elements, and p is a parameter governing how steep the decline in weights is: setting p to 0 would imply considering only the top element of the rank. In this setting, a low RBO score can be interpreted as indicating that senses that are closest to the contexts are different (thus suggesting that the sense intended by the polysemous term is different across texts), whilst the opposite case indicates that the senses more fitting to both contexts are same or similar, thereby authorizing to judge them as similar. For the task at hand, we simply assigned same sense when the RBO score exceeded a threshold set to 0.8.[9]

[9] The RBO parameter p has been optimized and set to .9, which is a setting also in accord with literature (Webber, Moffat, and Zobel, 2010).

Table 4.16: Results obtained by experimenting on the WiC dataset. Figures report the accuracy obtained for the three portions of the dataset and divided by POS.

System	Test	Training			Development		
		All	Nouns	Verbs	All	Nouns	Verbs
Contextualised word embeddings							
BERT-large[1]	68.4	-	-	-	-	-	-
WSD[2]	67.7	-	-	-	-	-	-
Ensemble[3]	66.7	-	-	-	-	-	
BERT-large[4]	65.5	-	-	-	-	-	-
ELMo-weighted[5]	61.2	-	-	-	-	-	-
Context2vec[4]	59.3	-	-	-	-	-	-
Elmo[4]	57.7	-	-	-	-	-	-
Sense representations							
DeConf[4]	58.7	-	-	-	-	-	-
SW2V[4]	58.1	-	-	-	-	-	-
JBT[4]	53.6	-	-	-	-	-	
LessLex (c-rnk-sim)	58.9	59.4	58.8	60.1	60.5	58.0	64.6
LessLex (RBO)	59.2	61.1	59.4	62.9	63.0	62.0	64.6
N2V (c-rnk-sim)	-	-	54.1	-	-	53.2	-
N2V (RBO)	-	-	60.7	-	-	63.4	-

[1] Wang et al. (2019)
[2] Loureiro and Jorge (2019)
[3] Soler, Apidianaki, and Allauzen (2019)
[4] Mancini et al. (2017)
[5] Ansell, Bravo-Marquez, and Pfahringer (2019)

RESULTS The results obtained experimenting on the WiC dataset are reported in Table 4.16.

Previous results show that this dataset is very challenging for embeddings that do not directly grasp contextual information. The results of systems participating to this task can then been arranged into three main classes: those adopting embeddings featured by contextualised word embeddings, those experimenting with embeddings endowed with sense representations, and those implementing sentence level baselines (Pilehvar and Camacho-Collados, 2019). Given that the dataset is balanced (that is, it comprises an equal number of cases where the meaning of the polysemous term is preserved/different across sentences), and the fact that the task is a binary classification one, the random baseline is 50% accuracy. Systems employing sense representations (directly comparing to ours) obtained up to 58.7% accuracy score (Pilehvar and Collier, 2016). On the other side, those employing contextualized word embeddings achieved accuracy ranging from 57.7% accuracy (ELMo 1024-d, from the first LSTM hidden state) to 68.4% accuracy (BERT 1024-d, 24 layers, 340M parameters) (Pilehvar and Camacho-Collados, 2019).

Our resource directly compares with multi-prototype, sense-oriented, embeddings, namely JBT (Pelevina et al., 2016), DeConf (Pilehvar and Collier, 2016), and SW2V (Mancini et al., 2017). In spite of the simplicity of both adopted approaches (c-rnk-sim and RBO), by employing LessLex vectors we obtained higher accuracy values than those reported for such comparable resources (listed as 'Sense representations' in Figure 4.16).

We also experimented with N2V (with both c-rank-sim and RBO metrics), whose results are reported for nouns on the training and development subsets.[10] For such partial results we found slightly higher accuracy than obtained with LessLex with the RBO metrics. Unfortunately, however, N2V results can be hardly compared to ours, since the experiments on the test-set were executed through the CodaLab Competitions framework.[11] In fact the design of the competition does not permit to separate the results for nouns and verbs, as the gold standard for the test set is not publicly available,[12] so that we were not able to directly experiment on the test-set to deepen comparisons.

[10] Parameters setting for NASARI2Vec: in the c-rnk-sim, α was set to 0.7, and the threshold to 0.8; in the RBO run, p was set to 0.9 and the threshold to 0.9.

[11] https://competitions.codalab.org/competitions/20010.

[12] As of mid August 2019.

4.3.2 LessLex *and Semantic Text Similarity*

The last downstream task that we selected for the evaluation of LessLex is the Semantic Text Similarity (STS). Here the goal is to compute a similarity score between two given portions of text. STS plays an important role in a plethora of applications such as information retrieval, text classification, question answering, topic detection, and as such it is helpful to evaluate to what extent LessLex vectors are suited to a downstream application.

4.3.2.1 *Experimental setup*

We provide our results on two datasets popular for this task: the STS benchmark, and the SemEval-2017 Task 1 dataset, both by Cer et al. (2017). The former dataset has been built by starting from the corpus of English SemEval STS shared task data (2012-2017). Sentence pairs in the SemEval-2017 dataset feature a varied cross-lingual and multilingual setting, deriving from the Stanford Natural Language for Inference (SNLI) (Bowman et al., 2015) except for one track (one of two Spanish-English cross-lingual tasks, referred to as Track 4b. spa-spa), whose linguistic material has been taken from the WMT 2014 quality estimation task by Bojar et al. (2014). The translations in this dataset are the following: Arabic (ara-ara), Arabic-English (ara-eng), Spanish (spa-spa), Spanish-English (spa-eng), Spanish-English (spa-eng), English (eng-eng), Turkish-English (tur-eng).

To assess our embeddings in this task, we used the implementation of the HCTI system, participating in the SemEval-2017 Task 1 (Shao, 2017), kindly made available by the author.[13] HCTI obtained the overall third place in that SemEval competition. The HCTI system — implemented by using Keras (Chollet et al., 2015) and Tensorflow (Abadi et al., 2016)—generates sentence embeddings with twin convolutional neural networks; these are then compared through the cosine similarity metrics, and element-wise difference with the resulting values is fed to additional layers to predict similarity labels. Namely, a Fully Connected Neural Network is used to transfer the semantic difference vector to a probability distribution over similarity scores. Two layers are employed herein, the first one using 300 units with *tanh* activation function; the second layer is charged to compute the (similarity label) probability distribution with 6 units combined with *softmax* activation function. While the original HCTI system employs GloVe vectors (Pennington, Socher, and Manning, 2014), we used LessLex vectors in our experimentation.

In order to actually compare only the employed vectors by leaving unaltered the rest of the HCTI system, we adopted the same parameter setting as available in the software bundle implementing the approach proposed in (Shao, 2017). We were basically able to reproduce the results of the paper, except for the hand-crafted features; however, based on experimental evidence, these did not seem to produce significant improvements in the system's accuracy.

We devised two simple strategies to choose the word-senses to be actually fed to the HCTI system. In the first case we built the context vector (as illustrated in Equation 4.2), and selected for each input term the sense closest to such vector. The same procedure has been run on both texts being compared for similarity. In the following we refer to

[13] http://tiny.cc/dstsaz.

this strategy as to *c-rank*. In the second case we selected for each input term the sense closest to the terminological vector, in the same spirit as in the first component of the ranked similarity (rnk-sim, Equation 4.1). In the following this strategy is referred to as *t-rank*. As mentioned, in the original experimentation two runs of the HCTI system were performed: one exploiting MT to translate all sentences into English, and another one with no MT, but performing a specific training on each track, depending on the involved languages Shao, 2017, p.132. Since we are primarily interested in comparing LessLex and GloVe vectors, rather than the quality of services for MT, we experimented in the condition with no MT. However, in this setting the GloVe vectors could not be directly used to deal with the cross-lingual tracks of the SemEval-2017 dataset. Specific retraining (although with no handcrafted features) was performed by the HCTI system using the GloVe vectors on the multilingual tracks. In experimenting with LessLex vectors, the HCTI system was trained only on the English STS benchmark dataset also to deal with the SemEval-2017 dataset: that is, no Machine Translation step nor any specific re-training was performed in experiments with LessLex vectors to deal with cross-lingual tracks.

4.3.2.2 *Results*

Results are reported in Table 4.17, where the correlation scores obtained by experimenting with LessLex and GloVe vectors are compared.

Let us start by considering the results obtained by experimenting on the STS benchmark. Here, when using LessLex embeddings we obtained figures similar to those obtained by the HCTI system using GLoVe vectors; namely, we observe that the choice of senses based on

Table 4.17: Results on the STS task. Top: results on the STS benchmark. Bottom: results on the SemEval-2017 dataset. Reported results are Pearson correlation indices, measuring the agreement with human annotated data. In particular, we compare the Pearson scores obtained by the HCTI system using LessLex and GloVe vectors. As regards as the runs with GloVe vectors, we report results with no hand-crafted features (no HF), and without machine translation (no MT)

STS Benchmark (English)

Track	HCTI + LessLex		HCTI + GloVe
	(t-rank)	(c-rank)	(no HF)
dev	.819	.823	.824
test	.772	.786	.783

SemEval 2017

Track	HCTI + LessLex		HCTI + GloVe
	(t-rank)	(c-rank)	(no MT)
1. ara-ara	.534	**.618**	.437
2. ara-eng	.310	.476	-
3. spa-spa	**.800**	.730	.671
4a. spa-eng	.576	.558	-
4b. spa-eng	.143	.009	-
5. eng-eng	.811	.708	**.816**
6. tur-eng	.400	.433	-

the overall context (c-rank) provides little improvements with respect to both GloVe vectors and to the t-rank strategy.

As regards as the seven tracks in the SemEval-2017 dataset, we can distinguish between results on multilingual and cross-lingual subsets of data. As regards as the former ones (that is, the ara-ara, spa-spa and eng-eng tracks), HCTI with LessLex obtained higher correlation scores than when using GloVe embeddings in two cases: +0.181 on the Arabic task, +0.129 on the Spanish task, and comparable results (−0.005) on the English track. We stress that no re-training was performed on LessLex vectors on languages different from English, so that the improvement obtained in the tracks 1 and 3 (ara-ara and spa-spa, respectively) is even more relevant. We interpret this achievement as stemming from the fact that LessLex vectors contain both conceptual and terminological descriptions: this seems also to explain the fact that the advantage obtained by employing LessLex vectors w.r.t. GloVe is more sensible for languages where the translation and/or re-training are less effective, such as pairs involving either the Arabic or Turkish language. Also, we note that using contextual information (c-rank strategy) to govern the selection of senses ensures comparable results to the t-rank strategy across settings (with the exception of track 4b, where the drop in the correlation is very prominent, in one order of magnitude). Finally, it is interesting to observe that in dealing with cross-lingual texts that involve arguably less-covered languages (i.e., in the tracks 2 and 6, ara-eng and tur-eng), the c-rank strategy produced better results than the t-rank strategy.

To summarize the results on the STS task, by plugging LessLex embeddings into a state-of-the-art system such as HCTI we obtained results that either improve or are comparable to more computationally intensive approaches involving either MT or re-training, necessary to use GLoVe vectors in a multilingual and cross-lingual setting. One distinguishing feature of our approach is that of hosting terminological and conceptual information in the same semantic space: experimental evidence seems to confirm it as helpful in reducing the need for further processing, and beneficial to map different languages onto such unified semantic space.

4.3.3 *Final discussion*

The experimentation on LessLex has taken into account overall eleven languages, from different linguistic lineages, such as Arabic, coming from the Semitic phylum; Basque, a language isolate (reminiscent of the languages spoken in southwestern Europe before Latin); English and German, two West Germanic languages; Farsi, that as an Indo-Iranian language can be ascribed to the set of Indo-European languages; Spanish and Portuguese, that are Western Romance languages in the Iberian-Romance branch; French, from the Gallo-Romance branch of Western Romance languages; Italian, also from the Romance lineage; Russian, from the eastern branch of the Slavic family of languages; Turkish, in the group of Altaic languages, featured by phenomena such as vowel harmony and agglutination.

We employed LessLex embeddings in order to cope with three tasks: *i)* the traditional semantic similarity task, where we experimented on six different datasets (RG-65, WS-Sim-353, SimLex-999, SimVerbs-3500, SemEval-2017 (Task 2) and Goikoetxea-2018); *ii)* the contextual semantic similarity task, where we experimented on two datasets, SCWS and WiC; *iii)* the STS task, where the STS Benchmark and the SemEval-2017 (Task 1) dataset were used for the experimentation.

In the first mentioned task (Section 4.2.2) our experiments show that in most cases LessLex results improve on those by all other competitors. As competitors all the principal embeddings were selected that allow to cope with multilingual tasks: ConceptNet Numberbatch, NASARI, JOINTChyb, SenseEmbed, and Nasari2Vec. Two different experimental conditions were considered (MSV and CbA, Table 4.12). Both views on results indicate that our approach outperforms the existing ones. To the best of our knowledge this is the most extensive experimentation ever performed on as many benchmarks, and including results for as many resources.

In dealing with the Contextual Similarity task (Section 4.3.1) we compared our results with those obtained by using NASARI2VEC, which also contains descriptions for both terms and nominal concepts in the same semantic space, and with results available in literature. The obtained figures show that despite not being tuned for this task, our approach improves on previous results on the SCWS dataset. On the WiC dataset, results obtained by experimenting with LessLex vectors overcome all those provided by directly comparable resources. Results obtained by state-of-the-art approaches (employing contextualized sense embeddings) in this task are about 9% above those currently achieved through sense embeddings.

As regards as the third task on Semantic Text Similarity (Section 4.3.2), we used our embeddings by feeding them to a Convolutional Neural Network in place of GloVe embeddings. The main outcome of this experiment is that while our results are comparable to those obtained by using GloVe for English tracks, they improve on the results obtained with GloVe in the cross-lingual setting, even though these are specifically retrained on the considered tracks.

In general, handling sense-embeddings involves some further processing to select senses for input terms, while with word-embeddings one can typically benefit from the direct mapping term-vector. Hence, the strategy employed to select senses is relevant when using LessLex embeddings. Also — though indirectly — subject to evaluation was the proposed similarity metrics of ranked-similarity; it basically relies on ranking sense vectors based on their distance from the terminological one. Ranked-similarity clearly outperforms the maximization of cosine similarity on LessLex embeddings. Besides, the contextual ranked-similarity (which was devised to deal with the contextual similarity task) showed to perform well, by taking into account information from the context vector rather than from the terminological one.

5

CONCLUSIONS

In this work we discussed the relevance that lexical resources assumed in the last decades in the Natural Language Processing research; we then illustrated the motivations that have brought to the development of two novel resources: COVER and LessLex.

In Chapter 3 we introduced COVER along with COVERAGE, the algorithm designed to built it. COVER puts together the lexicographic precision which is proper to WordNet and BabelNet with the rich common-sense knowledge that features ConceptNet. The obtained vectors capture conceptual information in a compact and cognitively sound fashion. We have also shown that COVER is suitable for building NLP applications, in the fields of conceptual categorization, abstractness computation, keywords extraction and conceptual similarity. We have reported the results of a thorough experimentation, which was carried out on the conceptual similarity task. Although other approaches presently achieve higher accuracy, the system employing COVER obtains competitive results, and additionally is able to build explanations of the traits determining the conceptual similarity.

In Chapter 4 we presented the LessLex vectors. Such vectors are built by re-arranging distributional descriptions around senses rather than terms. These have been tested on the word similarity task, on the contextual similarity task, and on the semantic text similarity task, providing good to outstanding results, on all datasets employed. Also importantly, we have outlined the relevance of LessLex vectors in the broader context of research in natural language with focus on senses and conceptual representation, mentioning that having co-located sense and term representations may be helpful to investigate some issues in an area at the intersection of general AI, Cognitive Science, Cognitive Psychology, Knowledge Representation and, of course, Computational Linguistics. In these settings distributed representation of senses may be employed, either to enable further research or to solve specific tasks.

Differently from most embedding approaches, LessLex enjoys the feature of adopting a unique semantic space for concepts and terms from different languages. Far from being an implementation feature, the adopted semantic space describes a cognitively plausible space, compatible with the cognitive mechanisms governing lexical access, which is in general featured by conceptual mediation (Marconi, 1997). Thanks to this peculiar attribute we are allowed to compare and unveil meaning connections between terms across different languages. Such capabilities can be useful in characterising subtle and elusive meaning shift *phenomena*, such as diachronic sense modeling (Hu, S. Li, and S. Liang, 2019) and conceptual misalignment, which is a well-known issue, e.g., in the context of automatic translation. This issue has been approached, for the translation of European laws, through the design of formal ontologies (Ajani et al., 2010).

We also proposed a novel similarity measure, the ranked-similarity. Such measure originates from a simple intuition: in computing conceptual similarity, scanning and comparing each and every sense available in some fine-grained sense inventory may be unnecessary and confusing. Instead, we rank senses using their distance from the term; top ranked senses are more relevant, so that the formula to compute ranked-similarity refines cosine similarity by adding a mechanism for filtering and clustering senses based on their salience.

The topic of conceptual abstractness requires a special mention, since the investigation on abstract concepts has recently emerged as central in the multidisciplinary debate between grounded views of cognition versus modal (or symbolic) views of cognition (Bolognesi and Steen, 2018). We have shown that acquiring vector descriptions for concepts (as opposed to terms) appears to be beneficial to investigate the conceptual abstractness/concreteness issue (Colla, Mensa, Porporato, et al., 2018; Hill, Korhonen, and Bentz, 2014; Mensa, Porporato, and Radicioni, 2018a), and its contribution to lexical competence (Marconi, 1997; Paivio, 1969). Also accounting for conceptual abstractness may be beneficial in diverse NLP tasks, like WSD (O. O. Kwong, 2008), the semantic processing of figurative uses of language (Neuman et al., 2013; Turney et al., 2011), automatic translation and simplification (Z. Zhu, Bernhard, and Gurevych, 2010), the processing of social tagging information (Benz et al., 2011), and many others, as well.

It is important to remark how LessLex and COVER constitute an effort to build complementary and yet interoperable knowledge that can be used hand-to-hand to tackle high level tasks. The explainability of COVER and the high coverage and usability of LessLex together with the fact that they share a common conceptual layer provided by the BabelNet sense inventory constitute the founding stone for the development of future applications focused not only on the mere performance but also giving priority to the transparency and intelligibility of the systems them self.

In the last few years many different novelties have been proposed: the rise of distributed representations, in combination with neural architectures (think, e.g, to convolutional and recurrent networks), have modified the NLP landscape, as well as some application areas. These, such as Question Answering and Automatic Translation, also have witnessed a sudden tremendous rebirth.

In this frame it is not simple to provide a clear and definitive outlook on the directions that our discipline will follow in the next future. We can envisage, however, some chief traits. First, the shift from static to contextual representations, to represent sentence and even paragraph embeddings. These models are aimed at grasping complex (such as syntactic and semantic) features associated to word usage, and also to learn how these features vary across linguistic contexts, like in modeling polysemy. One challenge will be extending these representations, e.g., to deal with multi and cross-linguistic extensions of such resources. A second main focus will be aimed, in our view, at exploring and devising models to deal with figurative language: this line of investigation may be intended as part of the former one. In a sense, figurative language can be intended as a set of tools that are used for delivering semantic content in a concise way, more concise than allowed by literal and plain language. Although this sort of language is very common (and deeply rooted in our way to conceptualize some types of meanings, such as those associated to some abstract domains such as time, ideas, and so forth), how meaning is actually conveyed in figurative expressions is partly unknown. Explaining such semantic phenomena will enable to deal with language in richer and more expressive fashion, closer to real language. Finally, next steps will involve combining and composing sense representations: different sorts of composition will be deepened, such as based on syntactic information (e.g., subcategorization frames) or based on different representation principles: in this view, one challenge will be in representing and recognizing events in their mutual relationships, and in being able to categorize events at different levels of granularity.

The work done all throughout my PhD course, summarized in this thesis, is to some extent connected to all these directions. The resources developed are committed to using sense-oriented representations, that allow to cope with all mentioned challenges by contributing to interpretative frameworks along with experimental tasks. The effort to link different knowledge sources (including lexicographic, common-sense, encyclopedic information and distributed representations) produced COVER and LessLex. These are interoperable resources that employ a unified naming convention to refer sense representations, and to deal with different semantic phenomena in a unified way.

Part III

APPENDIX

A

APPENDIX A

COVER ON WORD SIMILARITY: CBA CONDITION. In this section we report the results obtained by testing COVER on the semantic similarity task. Different from the results reported in Section 4.2.2.1, in this case only the fraction of each dataset covered by all considered resources was used for testing.

Table A.1: Results on the subset of the multilingual and cross-lingual RG-65 dataset containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

RG-65	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
[Word] eng [1.0]	.64	.59	**.91**	.86	**.91**	**.90**	.67	.67	.84	.86	.75	.81	.80	.75
[Sense] eng [1.0]	-	-	**.94**	**.91**	-	-	.81	.76	-	-	.72	.76	.78	.73
fas (N) [.69]	.78	.73	.86	.87	**.88**	**.89**	.71	.69	-	-	.72	.60	-	-
spa (N) [.98]	.82	.82	**.92**	**.93**	**.92**	**.93**	.91	.91	.80	.83	.82	.84	-	-
por-fas (N) [.81]	.73	.72	.91	**.90**	**.93**	.89	.79	.76	-	-	.76	.70	-	-
fra-por (N) [.97]	.83	.84	**.93**	**.89**	**.93**	**.89**	.76	.69	-	-	.81	.73	-	-
fra-fas (N) [.87]	.72	.72	.90	.88	**.93**	**.89**	.73	.69	-	-	.74	.68	-	-
fra-spa (N) [.99]	.81	.80	**.93**	**.91**	**.93**	.89	.85	.83	-	-	.88	.86	-	-
fra-deu (N) [.99]	.82	.86	**.91**	**.90**	.89	.88	.81	.78	-	-	.78	.76	-	-
spa-por (N) [.98]	.83	.83	**.93**	**.92**	**.93**	**.92**	.83	.81	-	-	.80	.79	-	-
spa-fas (N) [.82]	.71	.69	.92	**.92**	**.93**	.91	.83	.82	-	-	.78	.83	-	-
eng-por (N) [.99]	.74	.72	**.94**	**.90**	.92	**.90**	.79	.76	-	-	.80	.77	-	-
eng-fas (N) [.83]	.68	.61	.92	.89	**.93**	**.92**	.79	.74	-	-	.78	.74	-	-
eng-fra (N) [1.0]	.71	.70	**.94**	**.92**	.92	.91	.76	.73	-	-	.81	.75	-	-
eng-spa (N) [.99]	.73	.71	**.93**	**.93**	**.93**	.92	.85	.85	.84	.85	.80	.85	-	-
eng-deu (N) [.98]	.74	.72	**.92**	**.90**	.90	**.90**	.83	.81	-	-	.77	.80	-	-
deu-por (N) [.96]	.89	.86	**.93**	**.89**	.92	.88	.82	.78	-	-	.77	.74	-	-
deu-fas (N) [.81]	.76	.74	**.92**	**.91**	**.92**	.90	.88	.81	-	-	.82	.82	-	-
deu-spa (N) [.97]	.85	.86	**.92**	**.91**	.91	.90	.89	.86	-	-	.80	.81	-	-

Table A.2: Results on the subset of the WS-Sim-353 dataset containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

WS-Sim-353	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N) [.97]	.67	.65	**.78**	**.79**	**.78**	**.79**	.60	.61	.75	.76	.69	.73	.71	.70
ita (N) [.92]	.68	.69	.74	**.77**	**.75**	**.77**	.66	.65	.69	.70	.65	.71	-	-
deu (N) [.88]	.77	.74	.83	.81	**.84**	**.83**	.70	.69	-	-	.65	.64	-	-
rus (N) [.83]	.75	.76	.77	.78	**.79**	**.79**	.66	.66	-	-	.63	.64	-	-

Table A.3: Results on the subset of the SimVerbs-3500 dataset containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

SimVerbs-3500	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (V) [1.0]	.58	.56	**.67**	**.66**	.62	.60	-	-	.56	.56	.45	.42	.31	.30

Table A.4: Results on the subset of the multilingual SimLex-999 containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

SimLex-999	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N) [1.0]	.51	.52	**.69**	**.67**	.66	.63	.41	.39	.55	.53	.52	.49	.46	.44
eng (V) [1.0]	.62	.56	**.67**	**.65**	.61	.58	-	-	.51	.50	.54	.49	-	-
eng (A) [1.0]	**.84**	**.83**	.82	.79	.80	.78	-	-	.63	.62	.55	.51	-	-
eng (*) [1.0]	.57	.53	**.70**	**.69**	.67	.65	-	-	.55	.54	.53	.49	-	-
ita (N) [.96]	.50	.49	**.66**	**.64**	.64	.62	.48	.49	.48	.49	.56	.50	-	-
ita (V) [.96]	.58	.53	**.70**	**.63**	.69	.59	-	-	.57	.50	.56	.45	-	-
ita (A) [.95]	.68	.57	**.77**	**.70**	.73	.64	-	-	.40	.30	.61	.49	-	-
ita (*) [.96]	.49	.43	**.67**	**.63**	.65	.62	-	-	.48	.46	.55	.48	-	-
deu (N) [.94]	.58	.57	.66	.65	**.68**	**.66**	.46	.47	-	-	.48	.44	-	-
deu (V) [.73]	.56	.53	.63	**.60**	**.64**	.58	-	-	-	-	.51	.46	-	-
deu (A) [.67]	.74	.70	.76	.73	**.80**	**.75**	-	-	-	-	.50	.39	-	-
deu (*) [.86]	.59	.57	.66	.65	**.69**	**.67**	-	-	-	-	.47	.42	-	-
rus (N) [.86]	.45	.43	**.54**	**.51**	**.54**	.49	.23	.23	-	-	.26	.21	-	-
rus (V) [.20]	.60	.54	.58	.59	**.66**	**.60**	-	-	-	-	.42	.28	-	-
rus (A) [.06]	.92	.87	**.94**	**.91**	**.94**	.87	-	-	-	-	.62	.24	-	-
rus (*) [.63]	.46	.44	**.55**	**.51**	**.55**	.50	-	-	-	-	.27	.21	-	-

Table A.5: Results on the subset of the SemEval 17 Task 2 dataset containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

SemEval-2017	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N) [.66]	.70	.70	**.84**	**.86**	.83	.85	.57	.59	.75	.77	.71	.75	.73	.73
deu (N) [.73]	.78	.79	**.84**	.85	**.84**	**.86**	.68	.68	-	-	.67	.69	-	-
ita (N) [.61]	.73	.73	**.82**	**.84**	.80	.82	.75	.76	.76	.78	.71	.77	-	-
spa (N) [.62]	.77	.79	**.84**	**.86**	.81	.84	.70	.71	.78	.80	.73	.78	-	-
fas (N) [.34]	.69	.72	**.79**	**.82**	.75	.80	.58	.59	-	-	.65	.70	-	-
deu-spa (N) [.73]	.78	.80	**.84**	**.86**	.82	.84	.71	.72	-	-	.70	.74	-	-
deu-ita (N) [.74]	.77	.78	**.83**	**.85**	.82	.84	.72	.73	-	-	.69	.73	-	-
eng-deu (N) [.82]	.78	.79	**.85**	**.86**	.83	.85	.67	.68	-	-	.70	.72	-	-
eng-spa (N) [.63]	.74	.75	**.85**	**.87**	.83	.85	.65	.66	.75	.78	.72	.77	-	-
eng-ita (N) [.62]	.73	.74	**.85**	**.87**	.83	.85	.69	.70	.73	.75	.72	.77	-	-
spa-ita (N) [.61]	.75	.76	**.84**	**.86**	.81	.84	.74	.74	.70	.71	.72	.78	-	-
deu-fas (N) [.49]	.75	.78	**.84**	**.86**	.81	.85	.71	.72	-	-	.69	.74	-	-
spa-fas (N) [.49]	.72	.74	**.84**	**.86**	.80	.84	.70	.72	-	-	.70	.77	-	-
fas-ita (N) [.49]	.71	.72	**.81**	**.84**	.72	.82	.70	.72	-	-	.69	.75	-	-
eng-fas (N) [.54]	.70	.71	**.82**	**.85**	.79	.82	.65	.68	-	-	.70	.75	-	-

Table A.6: Results on the subset of the Goikoetxea dataset containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

Goikoetxea	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
spa-eus (N) [.75]	.75	.71	.80	**.74**	**.81**	.73	.74	.73	.69	.66	.74	.70	-	-
eng-eus (N) [.77]	.75	.72	**.93**	**.91**	**.93**	.90	.91	.90	.87	.84	.84	.86	-	-
eng-spa (N) [.99]	.73	.71	**.93**	**.93**	**.93**	.92	.85	.85	.84	.85	.80	.85	-	-
eus-ita (N) [.72]	.62	.66	**.69**	**.73**	.67	.63	.57	.59	.58	.63	.53	.56	-	-
spa-ita (N) [.93]	.60	.65	**.67**	**.75**	.66	.74	.58	.59	.56	.61	.53	.59	-	-
spa-eus (N) [.73]	.67	.70	**.74**	**.79**	.71	.78	.66	.67	.70	.74	.60	.64	-	-
eng-ita (N) [.96]	.59	.64	**.70**	.76	**.70**	**.77**	.51	.52	.61	.66	.51	.58	-	-
eng-eus (N) [.75]	.64	.67	**.75**	**.80**	.74	**.80**	.58	.60	.72	.76	.58	.63	-	-
eng-spa (N) [.97]	.62	.66	**.72**	**.78**	.71	**.78**	.55	.56	.68	.74	.57	.64	-	-
eng-spa (N) [.97]	.50	.49	**.67**	**.65**	.64	.62	.52	.51	.56	.52	.55	.52	-	-
eng-spa (V) [.96]	.53	.49	**.62**	**.60**	.59	.57	-	-	.48	.46	.53	.49	-	-
eng-spa (A) [.80]	.76	**.77**	**.77**	**.77**	**.77**	**.77**	-	-	.59	.60	.56	.50	-	-
eng-spa (*) [.95]	.54	.52	**.67**	**.66**	.65	.64	-	-	.54	.52	.55	.51	-	-
eng-ita (N) [.97]	.53	.53	**.71**	**.69**	.68	.66	.46	.47	.53	.51	.55	.52	-	-
eng-ita (V) [.58]	.62	.55	**.71**	**.67**	.67	.60	-	-	.51	.45	.56	.46	-	-
eng-ita (A) [.80]	.79	.73	**.84**	**.78**	.78	.70	-	-	.41	.36	.61	.48	-	-
eng-ita (*) [.82]	.56	.53	**.72**	**.70**	.69	.67	-	-	.50	.48	.56	.50	-	-
spa-ita (N) [.96]	.53	.53	**.68**	**.67**	.66	.65	.47	.49	.48	.47	.56	.54	-	-
spa-ita (V) [.56]	.56	.52	**.65**	**.60**	.64	.58	-	-	.47	.42	.56	.49	-	-
spa-ita (A) [.78]	.73	.66	**.79**	**.73**	.76	.69	-	-	.43	.38	.63	.51	-	-
spa-ita (*) [.80]	.55	.53	**.68**	**.66**	.67	.65	-	-	.47	.45	.56	.51	-	-

B

APPENDIX B

RESOURCES DOWNLOADS. The resources presented in this work can be found on the https://ls.di.unito.it website, and specifically:

- COCA senses (via CLOSEST) : https://ls.di.unito.it/resources/closest
- COVER and ABS-COVER: https://ls.di.unito.it/resources/cover
- LESSLEX: https://ls.di.unito.it/resources/lesslex
- Metaphor detection dataset: https://ls.di.unito.it/other-resources/metaphordetection

ALPHABETICAL INDEX

BIBLIOGRAPHY

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. (2016). "Tensorflow: A system for large-scale machine learning." In: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283 (cit. on p. 131).

Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa (2009). "A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches." In: *Proceedings of NAACL*. NAACL '09. Association for Computational Linguistics, pp. 19–27 (cit. on pp. 27, 49, 51, 52, 105).

Ajani, Gianmaria, Guido Boella, Leonardo Lesmo, Alessandro Mazzei, Daniele P. Radicioni, and Piercarlo Rossi (2010). "Multilevel legal ontologies." In: *Lecture Notes in Computer Science* 6036 LNAI, pp. 136–154. DOI: 10.1007/978-3-642-12837-0_8 (cit. on p. 140).

Andreas, Jacob and Dan Klein (2014). "How much do word embeddings encode about syntax?" In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2, pp. 822–827 (cit. on p. 4).

Ansell, Alan, Felipe Bravo-Marquez, and Bernhard Pfahringer (2019). "An ELMo-inspired approach to SemDeep-5's Word-in-Context task."

In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 2019* 10.2, pp. 62–66 (cit. on p. 128).

Arora, Sanjeev, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski (2018). "Linear algebraic structure of word senses, with applications to polysemy." In: *Transactions of the Association for Computational Linguistics* 6, pp. 483–495 (cit. on p. 123).

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre (2018). "Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations." In: *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 5012–5019 (cit. on p. 17).

Auer, Sören, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives (2007). "DBpedia: A nucleus for a web of open data." In: *The Semantic Web*. Springer, pp. 722–735 (cit. on p. 14).

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani (2010). "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining." In: *LREC*. Vol. 10. 2010, pp. 2200–2204 (cit. on p. 11).

Baker, Collin F, Charles J Fillmore, and John B Lowe (1998). "The Berkeley Framenet Project." In: *Proceedings of the 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pp. 86–90 (cit. on p. 7).

Bambini, Valentina, Donatella Resta, and Mirko Grimaldi (2014). "A dataset of metaphors from the Italian literature: exploring psycholinguistic variables and the role of context." In: *PloS one* 9.9, pp. 1–13 (cit. on p. 76).

Bansal, Mohit, Kevin Gimpel, and Karen Livescu (2014). "Tailoring continuous word representations for dependency parsing." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2, pp. 809–815 (cit. on p. 3).

Baroni, Marco, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta (2009). "The WaCky wide web: a collection of very large linguistically processed web-crawled corpora." In: *Language Resources and Evaluation* 43.3, pp. 209–226 (cit. on p. 84).

Baroni, Marco, Georgiana Dinu, and Germán Kruszewski (2014). "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 238–247 (cit. on p. 15).

El-Beltagy, Samhaa R and Ahmed Rafea (2009). "KP-Miner: A keyphrase extraction system for English and Arabic documents." In: *Information Systems* 34.1, pp. 132–144 (cit. on p. 71).

Benz, Dominik, Christian Körner, Andreas Hotho, Gerd Stumme, and Markus Strohmaier (2011). "One tag to bind them all: Measuring term abstractness in social metadata." In: *Proceedings of ESWC*. Springer, pp. 360–374 (cit. on pp. 77, 141).

Berant, Jonathan and Percy Liang (2014). "Semantic parsing via paraphrasing." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 1415–1425 (cit. on p. 4).

Birke, Julia and Anoop Sarkar (2006). “A clustering approach for nearly unsupervised recognition of nonliteral language.” In: *Proceedings of the 11th conference of EACL* (cit. on p. 77).

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2016). “Enriching Word Vectors with Subword Information.” In: *arXiv preprint arXiv:1607.04606* (cit. on p. 16).

Bojar, Ondrej, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. (2014). “Findings of the 2014 workshop on statistical machine translation.” In: *Proceedings of the ninth Workshop on Statistical Machine Translation*, pp. 12–58 (cit. on p. 130).

Bolognesi, Marianna and Gerard Steen (2018). “Editors’ Introduction: Abstract Concepts: Structure, Processing, and Modeling.” In: *Topics in Cognitive Science* 10.3, pp. 490–500 (cit. on p. 141).

Bosco, Cristina, Viviana Patti, and Andrea Bolioli (2013). “Developing corpora for sentiment analysis: The case of irony and senti-tut.” In: *IEEE Intelligent Systems* 28.2, pp. 55–63 (cit. on p. 3).

Bowman, Samuel R, Gabor Angeli, Christopher Potts, and Christopher D. Manning (2015). “A large annotated corpus for learning natural language inference.” In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642 (cit. on p. 130).

Brysbaert, Marc, Amy Beth Warriner, and Victor Kuperman (2014). “Concreteness ratings for 40,000 generally known English word lemmas." In: *BEHAV RES METH* 46.3, pp. 904–911 (cit. on pp. 82, 84).

Budanitsky, Alexander and Graeme Hirst (2006). "Evaluating WordNet-based Measures of Lexical Semantic Relatedness." In: *Computational Linguists* 32.1, pp. 13–47 (cit. on p. 26).

Camacho-Collados, José and Mohammad Taher Pilehvar (2018). "From word to sense embeddings: A survey on vector representations of meaning." In: *Journal of Artificial Intelligence Research* 63, pp. 743–788 (cit. on p. 20).

Camacho-Collados, José, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli (2017a). "SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity." In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*. Vancouver, Canada (cit. on pp. 49, 51, 60).

– (2017b). "Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity." In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 15–26 (cit. on pp. 102, 105, 106, 113).

Camacho-Collados, José, Mohammad Taher Pilehvar, and Roberto Navigli (2015a). "A framework for the construction of monolingual and cross-lingual word similarity datasets." In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Vol. 2, pp. 1–7 (cit. on pp. 104, 106).

Camacho-Collados, José, Mohammad Taher Pilehvar, and Roberto Navigli (2015b). “NASARI: a novel approach to a semantically-aware representation of items.” In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Tecnologies*, pp. 567–577 (cit. on pp. 4, 5, 51).

– (2016). “Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities.” In: *Artificial Intelligence* 240, pp. 36–64 (cit. on pp. 22, 51, 108).

Cambria, Erik, Bjorn Schuller, Bing Liu, Haixun Wang, and Catherine Havasi (2013). “Knowledge-based approaches to concept-level sentiment analysis.” In: *IEEE Intelligent Systems* 28.2, pp. 12–14 (cit. on p. 3).

Cambria, Erik, Robyn Speer, Catherine Havasi, and Amir Hussain (2010). “SenticNet: A Publicly Available Semantic Resource for Opinion Mining.” In: *AAAI fall symposium: commonsense knowledge*. Vol. 10 (cit. on p. 31).

Cer, Daniel, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia (2017). “SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation.” In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14 (cit. on p. 130).

Chandar, Sarath, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha (2014). “An autoencoder approach to learning bilingual word representations.” In:

Advances in Neural Information Processing Systems, pp. 1853–1861 (cit. on p. 17).

Changizi, Mark A (2008). “Economically organized hierarchies in WordNet and the Oxford English Dictionary.” In: *Cognitive Systems Research* 9, pp. 214–228 (cit. on p. 77).

Chen, Tao, Ruifeng Xu, Yulan He, and Xuan Wang (2015). “Improving distributed representation of word sense via wordnet gloss composition and context clustering.” In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 15–20 (cit. on pp. 21, 125).

Chen, Xinxiong, Zhiyuan Liu, and Maosong Sun (Jan. 2014). “A Unified Model for Word Sense Representation and Disambiguation.” In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1025–1035. DOI: 10.3115/v1/D14-1110 (cit. on pp. 21, 125).

Chi, Ta-Chung and Yun-Nung Chen (2018). “CLUSE: Cross-Lingual Unsupervised Sense Embeddings.” In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 271–281 (cit. on p. 124).

Chi, Ta-Chung, Ching-Yen Shih, and Yun-Nung Chen (2018). “BCWS: Bilingual Contextual Word Similarity.” In: *arXiv preprint arXiv:1810.08951* (cit. on p. 124).

Chiu, Billy, Anna Korhonen, and Sampo Pyysalo (2016). “Intrinsic evaluation of word vectors fails to predict extrinsic performance.” In:

Proceedings of the 1st workshop on evaluating vector-space representations for NLP, pp. 1–6 (cit. on p. 121).

Cho, Kyunghyun, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (2014). "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734 (cit. on p. 4).

Chollet, François et al. (2015). *Keras*. URL: https://github.com/keras-team/keras (cit. on p. 131).

Colla, Davide, Enrico Mensa, Aureliano Porporato, and Daniele P Radicioni (2018). "Conceptual Abstractness: from Nouns to Verbs." In: *5th Italian Conference on Computational Linguistics, CLiC-it 2018*. CEUR Workshop Proceedings, pp. 70–75 (cit. on pp. 76, 83, 141).

Colla, Davide, Enrico Mensa, and Daniele P. Radicioni (2017). "Semantic Measures for Keywords Extraction." In: *AI*IA 2017: Advances in Artificial Intelligence*. Lecture Notes for Artificial Intelligence. Springer (cit. on p. 66).

– (2020). "LessLex: Linking multilingual Embeddingsto SenSe representations of Lexical items." In: *Computational Linguistics (to appear)* 46.2 (cit. on p. 91).

Colla, Davide, Enrico Mensa, Daniele P. Radicioni, and Antonio Lieto (2018). "Tell Me Why: Computational Explanation of Conceptual Similarity Judgments." In: *Proceedings of the 17th International Conference on Information Processing and Management of Uncertainty in*

Knowledge-Based Systems (IPMU), Special Session on Advances on Explainable Artificial Intelligence. Communications in Computer and Information Science (CCIS). Cham: Springer International Publishing (cit. on p. 58).

Coltheart, Max (1981). “The MRC psycholinguistic database.” In: *The Quarterly Journal of Experimental Psychology Section A* 33.4, pp. 497–505 (cit. on p. 82).

Conneau, Alexis, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou (2018). “Word translation without parallel data.” In: *arXiv preprint arXiv:1710.04087* (cit. on p. 18).

Coulmance, Jocelyn, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum (2015). “Trans-gram, Fast Cross-lingual Word-embeddings.” In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1109–1113 (cit. on p. 18).

Davies, Mark (2009). “The 385+ million word Corpus of Contemporary American English (1990–2008+): Design, architecture, and linguistic insights.” In: *International Journal of Corpus Linguistics* 14.2, pp. 159–190 (cit. on pp. 34, 99).

Denecke, Kerstin (2008). “Using sentiwordnet for multilingual sentiment analysis.” In: *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on.* IEEE, pp. 507–512 (cit. on p. 3).

Deng, J., K. Li, M. Do, H. Su, and L. Fei-Fei (2009). “Construction and Analysis of a Large Scale Image Ontology.” In: Vision Sciences Society (cit. on p. 12).

Devitt, Ann and Khurshid Ahmad (2013). “Is there a language of sentiment? An analysis of lexical resources for sentiment analysis.” In: *Language Resources and Evaluation* 47.2, pp. 475–511 (cit. on p. 3).

Duong, Long, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn (2016). “Learning Crosslingual Word Embeddings without Bilingual Corpora.” In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1285–1295 (cit. on p. 17).

Evans, Jonathan St BT and Keith Ed Frankish (2009). *In two minds: Dual processes and beyond.* Oxford University Press (cit. on p. 65).

Faruqui, Manaal, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith (2014). “Retrofitting word vectors to semantic lexicons.” In: *arXiv preprint arXiv:1411.4166* (cit. on pp. 16, 19).

Faruqui, Manaal and Chris Dyer (2014). “Improving vector space word representations using multilingual correlation.” In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 462–471 (cit. on p. 17).

– (2015). “Non-distributional Word Vector Representations.” In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 464–469 (cit. on p. 119).

Finkelstein, Lev, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin (2001). “Placing search in context: The concept revisited.” In: *Proceedings of the 10th Interna-*

tional Conference on World Wide Web. ACM, pp. 406–414 (cit. on p. 49).

– (2002). "Placing search in context: The concept revisited." In: *ACM Transactions on information systems* 20.1, pp. 116–131 (cit. on p. 104).

Francopoulo, Gil, Nuria Bel, Monte George, Nicoletta Calzolari, Monica Monachini, Mandy Pet, and Claudia Soria (2009). "Multilingual resources for NLP in the lexical markup framework (LMF)." In: *Language Resources and Evaluation* 43.1, pp. 57–70 (cit. on p. 3).

Gerz, Daniela, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen (2016). "SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity." In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Jian Su, Xavier Carreras, and Kevin Duh. The Association for Computational Linguistics, pp. 2173–2182. ISBN: 978-1-945626-25-8 (cit. on pp. 105, 118).

Gînscă, Alexandru-Lucian, Emanuela Boroș, Adrian Iftene, Diana Trandabăț, Mihai Toader, Marius Corîci, Cenel-Augusto Perez, and Dan Cristea (June 2011). "Sentimatrix – Multilingual Sentiment Analysis Service." In: *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*. Portland, Oregon: Association for Computational Linguistics, pp. 189–195. URL: https://www.aclweb.org/anthology/W11-1725 (cit. on p. 3).

Goikoetxea, Josu, Aitor Soroa, and Eneko Agirre (June 2018). "Bilingual Embeddings with Random Walks over Multilingual Wordnets." In: *Knowledge-Based Systems* 150.C, pp. 218–230. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2018.03.017 (cit. on pp. 105, 108, 114, 115).

Gouws, Stephan, Yoshua Bengio, and Greg Corrado (2015). “BilBOWA: Fast Bilingual Distributed Representations without Word Alignments.” In: *Proceedings of The 32nd International Conference on Machine Learning*, pp. 748–756. URL: http://www.jmlr.org/proceedings/papers/v37/gouws15.pdf (cit. on pp. 4, 18).

Guo, Mandy, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernandez Abrego, Keith Stevens, Noah Constant, Yun-hsuan Sung, Brian Strope, et al. (2018). “Effective Parallel Corpus Mining using Bilingual Sentence Embeddings.” In: *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 165–176 (cit. on p. 18).

Harris, Zellig S (1954). “Distributional structure.” In: *Word* 10.2-3, pp. 146–162 (cit. on p. 15).

Hassan, Hany, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. (2018). “Achieving human parity on automatic chinese to english news translation.” In: *arXiv preprint arXiv:1803.05567* (cit. on p. 18).

Havasi, Catherine, Robyn Speer, and Jason Alonso (2007). “ConceptNet: A lexical resource for common sense knowledge.” In: *Recent advances in natural language processing V: selected papers from the International Conference Recent Advances in Natural Language Processing (RANLP)* 309, p. 269 (cit. on pp. 4, 7, 92).

Hill, Felix, Anna Korhonen, and Christian Bentz (2014). “A quantitative empirical analysis of the abstract/concrete distinction.” In: *Cognitive Science* 38.1, pp. 162–177 (cit. on p. 141).

Hill, Felix, Roi Reichart, and Anna Korhonen (2015). “Simlex-999: Evaluating semantic models with (genuine) similarity estimation.” In: *Computational Linguistics* 41.4, pp. 665–695 (cit. on pp. 105, 106).

Hisamoto, Sorami, Kevin Duh, and Yuji Matsumoto (2013). “An empirical investigation of word representations for parsing the web.” In: *Proceedings of the Association for Neuro Linguistic Programming*, pp. 188–193 (cit. on p. 4).

Hu, Renfen, Shen Li, and Shichen Liang (2019). “Diachronic Sense Modeling with Deep Contextualized Word Embeddings: An Ecological View.” In: *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pp. 3899–3908 (cit. on p. 140).

Huang, Eric H., Richard Socher, Christopher D. Manning, and Andrew Y. Ng (2012). “Improving word representations via global context and multiple word prototypes.” In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pp. 873–882 (cit. on pp. 20, 21, 106, 122, 123, 125).

Iacobacci, Ignacio, Mohammad Taher Pilehvar, and Roberto Navigli (2015). “Sensembed: Learning sense embeddings for word and relational similarity.” In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Vol. 1, pp. 95–105 (cit. on pp. 5, 7, 21, 108, 123).

Iliev, Rumen and Robert Axelrod (2017). “The Paradox of Abstraction: Precision Versus Concreteness.” In: *Journal of Psycholinguistic Research* 46.3, pp. 715–729 (cit. on p. 77).

Jean-Louis, Ludovic, Amal Zouaq, Michel Gagnon, and Faezeh Ensan (2014). “An assessment of online semantic annotators for the keyword extraction task.” In: *Pacific Rim International Conference on Artificial Intelligence*. Springer, pp. 548–560 (cit. on pp. 71–73, 75).

Jiang, Jay J and David W Conrath (1997). “Semantic similarity based on corpus statistics and lexical taxonomy.” In: *arXiv preprint cmp-lg/9709008* (cit. on p. 25).

Jimenez, Sergio, Claudia Becerra, Alexander Gelbukh, Av Juan Dios Bátiz, and Av Mendizábal (2013). “Softcardinality-core: Improving text overlap with distributional measures for semantic textual similarity.” In: *Proceedings of *SEM 2013*. Vol. 1, pp. 194–201 (cit. on p. 48).

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov (2016). “Bag of Tricks for Efficient Text Classification.” In: *arXiv preprint arXiv:1607.01759* (cit. on p. 16).

Kahneman, Daniel (2011). *Thinking, fast and slow*. Macmillan (cit. on p. 65).

Kenter, Tom and Maarten De Rijke (2015). “Short text similarity with word embeddings.” In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, pp. 1411–1420 (cit. on p. 3).

Krishnakumaran, Saisuresh and Xiaojin Zhu (2007). “Hunting elusive metaphors using lexical resources.” In: *Procs of the Workshop on Computational Approaches to Figurative Language*, pp. 13–20 (cit. on p. 87).

Kusner, Matt, Yu Sun, Nicholas Kolkin, and Kilian Weinberger (2015). "From word embeddings to document distances." In: *International Conference on Machine Learning*, pp. 957–966 (cit. on p. 3).

Kwong, Oi Yee (2008). "Sense abstractness, semantic activation, and word sense disambiguation." In: *International Journal of Speech Technology* 11.3-4, p. 135 (cit. on p. 77).

Kwong, Olivia OY (2008). "A preliminary study on the impact of lexical concreteness on word sense disambiguation." In: *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation*, pp. 235–244 (cit. on p. 141).

Lakoff, George, Jane Espenson, and Alan Schwartz (Oct. 1991). *The Master Metaphor List.* Tech. rep. University of California at Berkeley (cit. on p. 88).

Le, Quoc V. and Tomas Mikolov (2014). "Distributed representations of sentences and documents." In: *International Conference on Machine Learning*, pp. 1188–1196 (cit. on p. 3).

Leacock, Claudia, George A Miller, and Martin Chodorow (1998). "Using corpus statistics and WordNet relations for sense identification." In: *Computational Linguistics* 24.1, pp. 147–165 (cit. on p. 25).

Leviant, Ira and Roi Reichart (2015a). "Judgment language matters: Multilingual vector space models for judgment language aware lexical semantics." In: *CoRR, abs/1508.00106* (cit. on p. 105).

– (2015b). "Separated by an un-common language: Towards judgment language informed vector space modeling." In: *arXiv preprint arXiv:1508.00106* (cit. on p. 104).

Levin, Beth (1993). *English verb classes and alternations: A preliminary investigation.* University of Chicago press (cit. on p. 7).

Lieto, Antonio, Enrico Mensa, and Daniele P. Radicioni (2016a). "A Resource-Driven Approach for Anchoring Linguistic Resources to Conceptual Spaces." In: *Procs of the XV International Conference of the Italian Association for Artificial Intelligence.* Vol. 10037. LNAI. Springer, pp. 435–449. ISBN: 978-3-319-49129-5. DOI: 10.1007/978-3-319-49130-1 (cit. on p. 31).

– (2016b). "Taming Sense Sparsity: a Common-Sense Approach." In: *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian.* URL: http://ceur-ws.org/Vol-1749/paper31.pdf (cit. on p. 34).

Lieto, Antonio, Andrea Minieri, Alberto Piana, and Daniele P. Radicioni (2015). "A knowledge-based system for prototypical reasoning." In: *Connection Science* 27.2, pp. 137–152 (cit. on p. 64).

Lieto, Antonio, Daniele P. Radicioni, and Valentina Rho (July 2015). "A Common-Sense Conceptual Categorization System Integrating Heterogeneous Proxytypes and the Dual Process of Reasoning." In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).* Buenos Aires: AAAI Press, pp. 875–881 (cit. on p. 64).

– (2017). "Dual PECCS: A Cognitive System for Conceptual Representation and Categorization." In: *Journal of Experimental & Theoretical Artificial Intelligence* 29.2, pp. 433–452. DOI: 10.1080/0952813X.2016.1198934. eprint: http://dx.doi.org/10.1080/0952813X.

2016.1198934. URL: http://dx.doi.org/10.1080/0952813X.2016.1198934 (cit. on pp. 64, 66).

Lieto, Antonio, Daniele P. Radicioni, Valentina Rho, and Enrico Mensa (2017). “Towards a Unifying Framework for Conceptual Represention and Reasoning in Cognitive Systems.” In: *Intelligenza Artificiale* 11.2, pp. 139–153 (cit. on pp. 64, 66).

Liu, Hugo and Push Singh (2004). “ConceptNet—a practical commonsense reasoning tool-kit.” In: *BT technology journal* 22.4, pp. 211–226 (cit. on p. 12).

Loureiro, Daniel and Alipio Jorge (2019). “LIAAD at SemDeep-5 Challenge: Word-in-Context (WiC).” In: *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pp. 1–5 (cit. on p. 128).

Luong, Thang, Hieu Pham, and Christopher D. Manning (2015). “Bilingual word representations with monolingual quality in mind.” In: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 151–159 (cit. on pp. 4, 17).

Mancini, Massimiliano, José Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli (2017). “Embedding Words and Senses Together via Joint Knowledge-Enhanced Training.” In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 100–111 (cit. on pp. 128, 129).

Manning, Christopher D. (2015). “Computational linguistics and deep learning.” In: *Computational Linguistics* 41.4, pp. 701–707 (cit. on p. 3).

Marconi, Diego (1997). *Lexical competence.* MIT Press (cit. on pp. 140, 141).

Marujo, Luís, Anatole Gershman, Jaime Carbonell, Robert Frederking, and João P Neto (2012). "Supervised Topical Key Phrase Extraction of News Stories using Crowdsourcing, Light Filtering and Co-reference Normalization." In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pp. 399–403 (cit. on p. 71).

McCrae, John, Guadalupe Aguado-de-Cea, Paul Buitelaar, Philipp Cimiano, Thierry Declerck, Asunción Gómez-Pérez, Jorge Gracia, Laura Hollink, Elena Montiel-Ponsoda, Dennis Spohr, et al. (2012). "Interchanging lexical resources on the Semantic Web." In: *Language Resources and Evaluation* 46.4, pp. 701–719 (cit. on p. 3).

Mensa, Enrico, Aureliano Porporato, and Daniele P. Radicioni (2018a). "Annotating Concept Abstractness by Common-Sense Knowledge." In: *International Conference of the Italian Association for Artificial Intelligence.* Springer, pp. 415–428 (cit. on pp. 76, 82, 141).

– (2018b). "Grasping Metaphors: Lexical Semantics in Metaphor Analysis." In: *The Semantic Web: ESWC 2018 Satellite Events.* Ed. by Aldo Gangemi, Anna Lisa Gentile, Andrea Giovanni Nuzzolese, Sebastian Rudolph, Maria Maleshkova, Heiko Paulheim, Jeff Z Pan, and Mehwish Alam. Cham: Springer International Publishing, pp. 192–195. ISBN: 978-3-319-98192-5 (cit. on pp. 76, 86).

Mensa, Enrico, Daniele P. Radicioni, and Antonio Lieto (2017a). "MERALI at SemEval-2017 Task 2 Subtask 1: a Cognitively Inspired approach." In: *Proceedings of the International Workshop on Semantic*

Evaluation (SemEval 2017). Association for Computational Linguistics (cit. on p. 47).

– (Apr. 2017b). “TTCS$^{\mathcal{E}}$: a Vectorial Resource for Computing Conceptual Similarity.” In: *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*. Valencia, Spain: Association for Computational Linguistics, pp. 96–101. URL: http://www.aclweb.org/anthology/W17-1912 (cit. on p. 31).

– (2018). “COVER: a linguistic resource combining common sense and lexicographic information.” In: *Language Resources and Evaluation* 52.4, pp. 921–948 (cit. on p. 31).

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). “Efficient Estimation of Word Representations in Vector Space.” In: *CoRR* abs/1301.3781. URL: http://arxiv.org/abs/1301.3781 (cit. on p. 51).

Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever (2013). “Exploiting similarities among languages for machine translation.” In: *arXiv preprint arXiv:1309.4168* (cit. on p. 17).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). “Distributed representations of words and phrases and their compositionality.” In: *Advances in Neural Information Processing Systems*, pp. 3111–3119 (cit. on pp. 7, 15, 19).

Miller, George A (1995). “WordNet: a lexical database for English.” In: *Communications of the ACM* 38.11, pp. 39–41 (cit. on pp. 3, 7, 9).

Miller, George A and Walter G Charles (1991). "Contextual correlates of semantic similarity." In: *Language and Cognitive Processes* 6.1, pp. 1–28 (cit. on pp. 24, 49, 104).

Miller, George A and Christiane Fellbaum (2007). "WordNet then and now." In: *Language Resources and Evaluation* 41.2, pp. 209–214 (cit. on p. 3).

Mimno, David M., Hanna M. Wallach, Edmund M. Talley, Miriam Leenders, and Andrew McCallum (2011). "Optimizing Semantic Coherence in Topic Models." In: *EMNLP*. ACL, pp. 262–272. ISBN: 978-1-937284-11-4 (cit. on p. 68).

Minsky, Marvin (2000). "Commonsense-based interfaces." In: *Communications of the ACM* 43.8, pp. 66–73 (cit. on p. 31).

Mohammad, Saif M and Graeme Hirst (2012). "Distributional measures of semantic distance: A survey." In: *arXiv preprint arXiv:1203.1858* (cit. on p. 26).

Moro, Andrea, Francesco Cecconi, and Roberto Navigli (2014). "Multilingual word sense disambiguation and entity linking for everybody." In: *Proceedings of the 2014 International Conference on Posters & Demonstrations Track-Volume 1272*. CEUR-WS. org, pp. 25–28 (cit. on p. 3).

Moro, Andrea, Alessandro Raganato, and Roberto Navigli (2014). "Entity linking meets word sense disambiguation: a unified approach." In: *Transactions of the Association for Computational Linguistics* 2, pp. 231–244 (cit. on p. 21).

Mrkšić, Nikola, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young (2016). "Counter-fitting Word Vectors to Linguistic Constraints." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–148 (cit. on p. 119).

Mu, Jiaqi, Suma Bhat, and Pramod Viswanath (2017). "Geometry of Polysemy." In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. URL: https://openreview.net/forum?id=HJpfMIFll (cit. on p. 123).

Navigli, Roberto (2009). "Word sense disambiguation: A survey." In: *ACM Computing Surveys (CSUR)* 41.2, p. 10 (cit. on p. 3).

Navigli, Roberto and Simone Paolo Ponzetto (2010). "BabelNet: Building a very large multilingual semantic network." In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 216–225 (cit. on p. 3).

– (2012). "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network." In: *Artificial Intelligence* 193, pp. 217–250 (cit. on pp. 4, 7, 11, 95).

Neelakantan, Arvind, Jeevan Shankar, Alexandre Passos, and Andrew McCallum (2014). "Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing(EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter

Daelemans. ACL, pp. 1059–1069. ISBN: 978-1-937284-96-1 (cit. on pp. 21, 123).

Nelson, Douglas L, Cathy L McEvoy, and Thomas A Schreiber (2004). "The University of South Florida free association, rhyme, and word fragment norms." In: *Behavior Research Methods, Instruments, & Computers* 36.3, pp. 402–407 (cit. on p. 105).

Neuman, Yair, Dan Assaf, Yohai Cohen, Mark Last, Shlomo Argamon, Newton Howard, and Ophir Frieder (2013). "Metaphor identification in large texts corpora." In: *PloS one* 8.4, e62343 (cit. on pp. 77, 141).

Newman, David, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin (June 2010). "Evaluating Topic Models for Digital Libraries." In: *The ACM/IEEE Joint Conference on Digital Libraries (JCDL2010)*. Gold Coast, Australia: ACM (cit. on pp. 68, 69).

Paivio, Allan (1969). "Mental imagery in associative learning and memory." In: *Psychological review* 76.3, p. 241 (cit. on p. 141).

Palmer, Martha, Olga Babko-Malaya, and Hoa Trang Dang (2004). "Different Sense Granularities for Different Applications." In: *Proceedings of Workshop on Scalable Natural Language Understanding* (cit. on p. 34).

Pedersen, Ted, Satanjeev Banerjee, and Siddharth Patwardhan (2005). "Maximizing semantic relatedness to perform word sense disambiguation." In: *University of Minnesota supercomputing institute research report UMSI* 25, p. 2005 (cit. on p. 25).

Pelevina, Maria, Nikolay Arefiev, Chris Biemann, and Alexander Panchenko (2016). "Making Sense of Word Embeddings." In: *Proceedings of the*

1st Workshop on Representation Learning for NLP, pp. 174–183 (cit. on p. 129).

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). “Glove: Global vectors for word representation.” In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543 (cit. on pp. 15, 19, 131).

Peters, Matthew E, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). “Deep contextualized word representations.” In: *Proceedings of NAACL-HLT*, pp. 2227–2237 (cit. on p. 7).

Pianta, Emanuele, Luisa Bentivogli, and Christian Girardi (2002). “MultiWordNet: developing an aligned multilingual database.” In: *First International Conference on Global WordNet*, pp. 293–302 (cit. on p. 11).

Pilehvar, Mohammad Taher and José Camacho-Collados (2019). “WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations.” In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1267–1273 (cit. on pp. 122, 128, 129).

Pilehvar, Mohammad Taher and Nigel Collier (2016). “De-Conflated Semantic Representations.” In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1680–1690 (cit. on p. 129).

Pilehvar, Mohammad Taher and Roberto Navigli (2015). “From senses to texts: An all-in-one graph-based approach for measuring semantic similarity.” In: *Artificial Intelligence* 228, pp. 95–128 (cit. on pp. 26, 50–52).

Reisinger, Joseph and Raymond J Mooney (2010). “Multi-prototype vector-space models of word meaning.” In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 109–117 (cit. on pp. 20, 21, 125).

Resnik, Philip (1995). “Using information content to evaluate semantic similarity in a taxonomy.” In: *arXiv preprint cmp-lg/9511007* (cit. on p. 24).

– (1998). “Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language.” In: *Journal of Artificial Intelligence Research* 11.1 (cit. on p. 25).

Richardson, Ray, Alan F Smeaton, and John Murphy (1994). “Using WordNet as a knowledge base for measuring semantic similarity between words.” In: *Proceedings of AICS conference*, pp. 1–15 (cit. on p. 24).

Robyn, Speer and Joanna Lowry-Duda (2017). “ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge.” In: *CoRR* abs/1704.03560. URL: http://arxiv.org/abs/1704.03560 (cit. on pp. 51, 52, 55).

Röder, Michael, Andreas Both, and Alexander Hinneburg (2015). “Exploring the space of topic coherence measures.” In: *Proceedings of the eighth ACM International Conference on Web Search and Data Mining*, pp. 399–408 (cit. on p. 69).

Rosch, Eleanor (1975). “Cognitive Representations of Semantic Categories.” In: *Journal of Experimental Psychology: General* 104.3, pp. 192–233 (cit. on p. 31).

Rubenstein, Herbert and John B Goodenough (1965). “Contextual correlates of synonymy.” In: *Communications of the ACM* 8.10, pp. 627–633 (cit. on pp. 49, 104).

Ruder, Sebastian, Ivan Vulić, and Anders Søgaard (2019). “A survey of cross-lingual word embedding models.” In: *Journal of Artificial Intelligence Research* 65, pp. 569–631 (cit. on p. 16).

Schwartz, Hansen A and Fernando Gomez (2008). “Acquiring knowledge from the web to be used as selectors for noun sense disambiguation.” In: *Proceedings of the Twelfth Conference on Computational Natural Language Learning*. ACL, pp. 105–112 (cit. on p. 25).

Schwartz, Hansen Andrew and Fernando Gomez (2011). “Evaluating semantic metrics on tasks of concept similarity.” In: *Twenty-Fourth International FLAIRS Conference*, p. 324 (cit. on pp. 50, 104, 110).

Schwenk, Holger and Matthijs Douze (2017). “Learning Joint Multilingual Sentence Representations with Neural Machine Translation.” In: *ACL 2017*, pp. 157–167 (cit. on p. 18).

Shao, Yang (2017). “HCTI at SemEval-2017 Task 1: Use convolutional neural network to evaluate semantic textual similarity.” In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 130–133 (cit. on pp. 130–132).

Soler, Aina Garí, Marianna Apidianaki, and Alexandre Allauzen (2019). “LIMSI-MULTISEM at the IJCAI SemDeep-5 WiC Challenge: Context Representations for Word Usage Similarity Estimation." In: *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pp. 6–11 (cit. on p. 128).

Speer, Robert and Joshua Chin (2016). "An ensemble method to produce high-quality word embeddings." In: *arXiv preprint arXiv:1604.01692* (cit. on p. 19).

Speer, Robyn, Joshua Chin, and Catherine Havasi (2017). "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge." In: pp. 4444–4451. URL: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972 (cit. on pp. 5, 7, 19, 51, 52, 108, 113).

Speer, Robyn and Joanna Lowry-Duda (2017). "ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge." In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 85–89. DOI: 10.18653/v1/S17-2008. URL: http://aclweb.org/anthology/S17-2008 (cit. on p. 91).

Stevens, Keith, Philip Kegelmeyer, David Andrzejewski, and David Buttler (2012). "Exploring topic coherence over many models and many topics." In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, pp. 952–961 (cit. on pp. 69, 70).

Tang, Duyu, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin (2014). "Learning sentiment-specific word embedding for twitter sentiment classification." In: *Proceedings of the 52nd Annual Meeting* *of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 1555–1565 (cit. on p. 4).

Theijssen, DL, H van Halteren, LWJ Boves, and NHJ Oostdijk (2011). "On the difficulty of making concreteness concrete." In: *CLIN Journal*, pp. 61–77 (cit. on pp. 78, 82).

Turney, Peter D, Yair Neuman, Dan Assaf, and Yohai Cohen (2011). "Literal and metaphorical sense identification through concrete and abstract context." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 680–690 (cit. on p. 141).

Tversky, Amos (1977). "Features of similarity." In: *Psychological review* 84.4, p. 327 (cit. on p. 48).

Vigliocco, Gabriella, Lotte Meteyard, Mark Andrews, and Stavroula Kousta (2009). "Toward a theory of semantic representation." In: *Language and Cognition* 1.2, pp. 219–247 (cit. on p. 77).

Vossen, Piek and Christiane Fellbaum (2009). "Multilingual FrameNets in Computational Lexicography: Methods and Applications." In: Trends in linguistics / Studies and monographs: Studies and monographs. Mouton de Gruyter. Chap. Universals and idiosyncrasies in multilingual WordNets (cit. on p. 34).

Vrandečić, Denny and Markus Krötzsch (2014). "Wikidata: a free collaborative knowledge base." In: (cit. on p. 12).

Vulić, Ivan and Anna Korhonen (2016). "On the role of seed lexicons in learning bilingual word embeddings." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 247–257 (cit. on p. 16).

Vulić, Ivan and Marie-Francine Moens (2015). "Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings." In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, pp. 363–372 (cit. on p. 17).

Wang, Alex, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2019). "Superglue: A stickier benchmark for general-purpose language understanding systems." In: *Advances in Neural Information Processing Systems*, pp. 3261–3275 (cit. on p. 128).

Webber, William, Alistair Moffat, and Justin Zobel (2010). "A similarity measure for indefinite rankings." In: *ACM Transactions on Information Systems (TOIS)* 28.4, p. 20 (cit. on pp. 126, 127).

Witten, Ian H, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning (1999). "KEA: practical automatic keyphrase extraction." In: *Proceedings of the fourth ACM conference on Digital libraries*, pp. 254–255 (cit. on p. 71).

Wu, Zhibiao and Martha Palmer (1994). "Verbs semantics and lexical selection." In: *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics.* ACL, pp. 133–138 (cit. on pp. 24, 25).

Xing, Xing, Yi Zhang, and Mei Han (2010). "Query difficulty prediction for contextual image retrieval." In: *European Conference on Information Retrieval*, pp. 581–585 (cit. on pp. 77, 82).

Yarlett, D and M Ramscar (2008). "Language learning through similarity-based generalization." In: *Unpublished PhD Thesis, Stanford University* (cit. on p. 15).

Zhu, Zhemin, Delphine Bernhard, and Iryna Gurevych (2010). “A monolingual tree-based translation model for sentence simplification.” In: *Proceedings of the 23rd International Conference on Computational Linguistics*. ACL, pp. 1353–1361 (cit. on pp. 77, 141).

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/

Thank you very much for your feedback and contribution.

Final Version as of July 16, 2020 (1.0).