Procesamiento del Lenguaje Natural, Revista nº 57, septiembre de 2016, págs. 109-116 recibido 20-03-2016 revisado 20-04-2016 aceptado 01-05-2016

Comparing Distributional Semantics Models for identifying groups of semantically related words

Comparación de dos modelos de semántica distribucional para identificar grupos de palabras semánticamente relacionadas

Venelin Kovatchev, Maria Salamó, M. Antònia Martí
Universitat de Barcelona
Gran Via 585, 08007 Barcelona, Spain
{vkovatchev, maria.salamo, amarti}@ub.edu

Abstract: Distributional Semantic Models (DSM) are growing in popularity in Computational Linguistics. DSM use corpora of language use to automatically induce formal representations of word meaning. This article focuses on one of the applications of DSM: identifying groups of semantically related words. We compare two models for obtaining formal representations: a well known approach (CLUTO) and a more recently introduced one (Word2Vec). We compare the two models with respect to the PoS coherence and the semantic relatedness of the words within the obtained groups. We also proposed a way to improve the results obtained by Word2Vec through corpus preprocessing. The results show that: a) CLUTO outperforms Word2Vec in both criteria for corpora of medium size; b) The preprocessing largely improves the results for Word2Vec with respect to both criteria.
Keywords: DSM, Word2Vec, CLUTO, semantic grouping

Resumen: Los Modelos de Semántica Distribucional (MSD) están siendo utilizados de manera extensiva en el área de la Lingüística Computacional. Los MSD utilizan corpus de uso de la lengua para inducir de manera automática diferentes tipos de representaciones sobre el significado de las palabras. Este artículo se centra en una de las aplicaciones de los MSD: la identificación de grupos de palabras semánticamente relacionadas. Se comparan dos modelos de obtención de representaciones formales: CLUTO, una herramienta estándar de clusterización y Word2Vec, una aproximación reciente al tema. Comparamos los resultados obtenidos con ambos modelos basándonos en dos criterios: la coherencia que presentan estas agrupaciones respecto de la categoría morfosintáctica y la cohesion semántica entre las palabras dentro de cada grupo. Se propone también como mejorar los resultados obtenidos con Word2Vec mediante su preprocesamiento morfosintáctico. Los resultados obtenidos demuestran que: a) CLUTO supera a Word2Vec en ambos criterios cuando se trata de corpus de tamaño medio: b) el preprocesamiento mejora de manera clara los resultados obtenidos con Word2Vec para ambos criterios.
Palabras clave: DSM, Word2Vec, CLUTO, agrupación semántica de palabras.

1 Introduction

In recent years, the availability of large corpora and the constantly increasing computational power of the modern computers have led to a growing interest in linguistic approaches that are automated and data-driven (Arppe et al., 2010). Distributional semantic models (DSM) (Turney and Pantel, 2010; Baroni and Lenci, 2010) and the vector representations (VR) they generate fit very well within this framework: the process of extracting vector representations is mostly automated and the content of the representations is data-driven.

The format of the vector is suitable for carrying out different mathematical manipulations. Vectors can be compared directly through an objective mathematical function. They can also be used as a dataset for various Machine Learning algorithms. VR are more often used on tasks related to lexical simi-

ISSN 1135-5948

larity and relational similarity (Turney and Pantel, 2010). In such tasks, the emphasis is on pairwise comparisons between vectors.

This article focuses on another use of the Vector Representations: the grouping of vectors, based on their similarity in the Distributional space. This grouping can be used, among other things, as a methodology for identifying groups of semantically related words. High quality groupings can serve for many purposes: they are a semantic resource on their own, but can also be applied for syntactic disambiguation or pattern identification and generation (Martí et al., Submitted, 2016), for example.

We compare two different methodologies for obtaining groupings of semantically related words in English - a well known approach (CLUTO) and a more recently introduced one (Word2Vec). The two methodologies are evaluated in terms of the quality of the obtained groups. We consider two criteria: 1) the semantic relatedness between the words in the group; and 2) the PoS coherence of the group. We evaluate the role of the corpus size with both methodologies and in the case of Word2Vec, the role of the linguistic preprocessing (lemmatization and PoS tagging).

The rest of this paper is organized as follows: Section 2 presents the general framework and related work. Section 3 describes the available data and tools. Section 4 presents the experiments and the results obtained. Finally Section 5 gives conclusions and identifies directions for future work.

2 Related work

Distributional Semantics Models (DSM) are based on the Distributional Hypothesis, which states that the meaning of a word can be represented in terms of the contexts in which it appears (Harris, 1954; Firth, 1957). As opposed to semantic approaches based on primitives (Boleda and Erk, 2015), approaches based on distributional semantics can obtain formal representations of word meaning from actual linguistic productions. Additionally, this data-driven process for semantic representation can mostly be automated.

Within the framework of DSM, one of the most common ways to formalize the word meaning is a vector in a multi-dimensional distributional space (Lenci, 2008). For this purpose, a matrix with size **m** by **n** is extracted from the corpus, representing the distribution of **m** words over **n** contexts. The format of a vector allows for direct quantitative comparison between words using the apparatus of linear algebra. At the same time it is a format preferred by many Machine Learning algorithms.

The choice of the matrix is central for the implementation of a particular DSM. Turney and Pantel (2010) suggest a classification of the DSM based on the matrix used. They analyze three different matrices: term-document, word-context, and pair-pattern. The different matrices represent different types of relations in the corpus and the choice of the matrix depends on the goals of the particular research.

Baroni and Lenci (2010) present a different, sophisticated approach for extracting information from the corpus. They organize the information as a third order tensor, with the dimensions representing <'word', 'link', 'word' >. This third order tensor can then be used to generate different matrices, without the need of going back to the original corpus.

In this paper we focus on one of the classical vector representations - the one based on word-context relation. It measures what Turney and Pantel (2010) call "attributional similarity". In particular, we are interested in the possibility to group vectors together, based on their relations in the distributional space.

Erk (2012) offers a survey of possible applications of different DSM. She lists clustering as an approach that can be used with vectors, for word sense disambiguation. Moisl (2015) presents a theoretical analysis on the usage of clustering in computational linguistics and identifies key aspects of the mathematical and linguistic argumentation behind it.

Here we analyze and compare two approaches that induce vector representations from a corpus and apply algorithms to identify sets of semantically related words. We are interested in the quality of the obtained groups, as we believe that they can be a useful, empirical, linguistic resource.

Martí et al. (Submitted, 2016) present a methodology named DISCOveR for identifying candidates to be constructions from a corpus. As part of this methodology they use CLUTO (Karypis, 2002) for clustering words based on their vector representations. Their

approach uses a word-context matrix where the context is defined by combining a syntactic dependency with a lemma. After all the vectors are extracted, CLUTO is used in order to obtain clusters of semantically related words. Later on these clusters are used to generate a list of the candidates to be constructions.

Mikolov et al. (2013) suggest a different approach towards extracting vector representations and grouping. Their methodology is based on deep learning and is intended for quick processing of very large corpora. Word2Vec[1], the tool they present, includes an integrated algorithm for grouping words based on proximity in space. The context they use for vector extraction is simple co-occurrence within a specified window of tokens. Originally, they make no use of linguistic preprocessing such as lemmatization, part of speech tagging or syntactic tagging. As part of this paper we evaluate the effect of linguistic preprocessing on the obtained vectors and groups.

3 Data and tools

In this section we present the corpus that we use in the evaluation (Section 3.1) and the two methodologies (Section 3.2 and Section 3.3).

3.1 The corpus

For all of the experiments described in this paper, we use PukWaC (Baroni et al., 2009)[2]. It is a 2 billion word corpus of English, built up from sites in the .uk domain. It is available online and is already preprocessed: XML tags and other non-linguistic information have been removed, it is lemmatized, PoS tagged and syntactically parsed. The PoS tagset is an extended version of the Penn Treebank tagset. The syntactic dependencies follow the CONLL-2008 shared task format.

3.2 Grouping with CLUTO

DISCOveR (Martí et al., Submitted, 2016) is a methodology for identifying candidates to be construction from a corpus. It uses vector representations, extracted from a corpus. CLUTO (Karypis, 2002) is used on these representations in order to obtain clusters of semantically related words. CLUTO is a software package for clustering low and high dimensional data sets and for analysis of the characteristics of the various clusters. CLUTO provides three different classes of clustering algorithms, based on partitional, agglomerative and graph-partitioning paradigms. It computes clustering solution based on one of the different approaches.

For this article, we are interested only in the first three steps of the DISCOveR process. Step 1 is the linguistic preprocessing of the corpus. The raw text is cleared from non-linguistic data, it is PoS tagged and syntactically parsed. In Step 2, the DSM matrix is constructed. The rows of the matrix correspond to lemmas and the columns correspond to contexts. Contexts in this approach are defined as a triple of syntactic relation, direction of the relation and lemma in [direction:relation:lemma] format[3]. This matrix is used to generate vector representations for the 10,000 most frequent words in the corpus. Next, Step 3 uses CLUTO to create clusters of semantically related lemmas from the DSM matrix and the corresponding vectors. The clusters are created based on shared contexts.

Martí et al. (Submitted, 2016) start from a raw, unprocessed corpus and in Step 1 they clear the corpus and tag it with the linguistic data relevant to the matrix extraction. The format they use is shown in Table 1.

Token	sanitarios
Lemma	sanitario
PoS	NCMP
Short PoS	n
Sent ID	000
Token ID	0
Dep ID	2
Dep Type	suj

Tabla 1: Diana-Araknion Format

The original DISCOveR experiment is done with the Diana-Araknion corpus of Spanish. For the purpose of this article, we replicated the process for English, using the Puk-

[1] Available at: https://code.google.com/archive/p/word2vec/

[2] Available at: http://wacky.sslmit.unibo.it

[3] For example, from the sentence "El barbero afeita la larga barba de Jaime", three different contexts of the noun lemma barba are generated: [<:dobj:afeitar_v], [>:mod:largo_a] and [>:de_sp:pn_n]. The example is from (Martí et al., Submitted, 2016)

WaC corpus. For step 1 we had to make sure that our preprocessing is equivalent to the one of Diana-Araknion. The corpus PukWaC is already preprocessed and the format is similar to the one of Diana-Araknion. However, in order to make it fully compatible, we had to make several modifications of the format and linguistic decisions. Regarding the format, we removed any remaining XML tags, enumerated the sentences in the corpus, and generated "short PoS"[4]. From the linguistic side, we had to decide whether all PoS and Dependencies were relevant for the vector generation or some of them could be merged together or even discarded in order to optimize and speed up the process.

The process of generating vectors and clusters is based on analyzing the contexts where each word appears in. A word is identified by its lemma and its PoS tag. However, in the PukWac tagset there are many PoS tags which specify not only the PoS of the token, but also contain information about other grammatical features, such as person, number, and tense. If these tags are kept unchanged, a separate vector will be generated for different forms of the same word, based on different PoS tag. To avoid this problem and to generate only one vector for all of the different word forms, we have decided to merge certain PoS tags under one category.

We decided to simplify the POS tagset further. It is a common practice in DSM to focus the experiment on the relations between content words. Function words and punctuation are usually not considered relevant contexts. Because of that, we have put them under the common tag "other". All of the changes on the PoS tagset are summarized in Table 2.

The list of syntactic dependencies in PukWaC is also not fully relevant to the task of vector generation. While the unnecessary PoS tags may lead to multiple vectors for the same word, unnecessary dependencies generate additional contexts, increasing the dimensionality of the vectors and leading to a more complicated computational process. Therefore the modification of the dependencies is mostly related to the optimization of the computational process. After analyzing the tagset, we have decided to merge the

[4]short PoS is a one letter tag representing the generic PoS tag of the lemma. In this experiment, short PoS is the first letter of the full PoS

Tag	Original tag	Description
J	JJ JJR JJS	Adjective
M	MD	Modal verb
N	NN NNS	Noun (common)
NP	NP NPS	Noun (personal)
R	RB RBR RBS RP	Adverb
S	IN	Preposition
V	VB* VH* VV*	Verb (all)
O	CC CD DT PDT EX FW LS POS PP* SYM TO UH W* punctuation	Rest

Tabla 2: PoS tagset modifications

OBJ and **IOBJ** tags due to some inconsistencies of their usage. We have also decided to discard the following relations: **CC** (conjunction), **CLF** (be/have in a complex tense), **COORD** (coordination), **DEP** (unclassified relation), **EXP** (experiencer in few very specific cases), **P** (punctuation), **PRN** (parenthetical), **PRT** (particle), **ROOT** (root clause).The final list of dependencies is shown in Table 3.

Dependency	Description
ADV	Unclassified adv
AMOD	Modifier of adj or adv
LGS	Logical subj
NMOD	Modifier of nom
OBJ	Direct or indirect obj
PMOD	Preposition
PRD	Predicative compl
SBJ	Subject
VC	Verb chain
VMOD	Modifier of verb
empty	No dependency

Tabla 3: Syntactic Dependencies

Once the corpus is preprocessed, the process of matrix extraction is mostly automated. For the matrix, we have only generated vectors for words that appear at least 5 times in the corpus. Out of them we have used only the vectors of the 10,000 most frequent words for the clustering process.

For the clustering process, we configure CLUTO to use direct clustering, based on the H2 criterion function, with 25 features

per cluster. We have ran the clusterization multiple times, ranging from 100 to 1,000 clusters. We then used CLUTO's H2 metric to determine the optimal number of clusters, which has been 800 for all of the experiments.

3.3 Grouping with Word2Vec

Word2Vec is based on the methodology proposed by Mikolov et al. (2013). It takes a raw corpus and a set of parameters and generates vectors and groups. The algorithm of Word2Vec is based on a two layer neural network that are trained to reconstruct linguistic context of words. Word2Vec includes two different algorithms - Continuous Bag-of-Words (CBOW) and Skip-Gram. CBOW learns representations based on the context as a whole - all of the words that co-occur with the target word in a specific window. Skip-Gram learns representation based on each single other word within a specified window. When using Word2Vec usually the emphasis is put on the choice of the paremeters for the algorithm, and not on the specifications of corpus. However, we consider that the specifications of the corpus (size and linguistic preprocessing) can largely affect the quality of the obtained results.

By default Word2Vec works with a raw corpus. Neither of the two models makes explicit use of morpho-syntactic information. However, by modifying the corpus, some morphological information can be used implicitly. If the token is replaced by its corresponding lemma or by the lemma and part of speech tag in a "lemma_pos" format, the resulting vectors would be different: using the lemma would generate only one vector for the word as opposed to separate vector for every word form; using PoS can make a distinction between homonyms with same spelling and different PoS. As part of our work we wanted to examine how linguistic preprocessing can affect the quality of the vectors. For that reason we created three separate corpus samples - one raw corpus, one where each token was replaced by its lemma, and one where each token was replaced by "lemma_pos". We generated vectors separately for each of the corpora. Unfortunately, there was no trivial way to introduce syntactic information implicitly in the models of Word2Vec.

4 Experiments

In this section we present the setup for the different experiments (Section 4.1), the evaluation criteria (Section 4.2), and the obtained results (Section 4.3).

4.1 Setup

We carried out a total of 15 experiments - 3 experiments using CLUTO and 12 experiments using Word2Vec. For the experiments with CLUTO, the only variation between the experiments was the size of the corpus: 4M tokens, 20M tokens, and 40M tokens[5]. In all the experiments we used the preprocessing described at Section 3.2, we generated vectors for the 10,000 most frequent words and we split them into 800 clusters. For the experiments with Word2Vec, we changed three parameters of the experiments: (1) the algorithm (CBOW and Skip-Gram), (2) the linguistic preprocessing of the corpus (raw, lemma, lemma and PoS), and (3) the size of the corpus (4M, 20M, and 40M). We carried out 9 experiments with CBOW (all size and preprocessing combinations) and 3 experiments with Skip-Gram (the three variants of the 40M corpus). Mikolov et al. (2013) identify two important parameters to be set up when using Word2Vec: the vector size and the window size. For the window size, we used 8, which is the recommended value. For the vector size, Mikolov et al. (2013) show that increasing vector size from 100 to 300 leads to significant improvement of the results, however further increase does not have big impact. For that reason we have chosen vector size of 400, which is above the recommended minimum. For the number of groups we used 800: the same number that was determined optimal for CLUTO. For the number of lemmas, we used the 10,000 most frequent ones, the same setup as with CLUTO.

4.2 Evaluation

The two methodologies and all of the different setups are evaluated based on the quality of the obtained groups. We consider two criteria: 1) The semantic relatedness between the words in each group; and 2) The PoS coherence of the groups. The PoS coherence is a secondary criterion which should be

[5]The 40M corpus contains in itself the 20M corpus. The 20M corpus contains in itself the 4M corpus. The same corpora has been used for the experiments with both CLUTO and with Word2Vec.

considered in addition to the semantic relatedness. Our intuition is that groups that are semantically related and PoS coherent are a better resource than groups that are only semantically related. For evaluating the semantic relations of the words in the groups, we present two methodologies - an automated method based on WordNet distances and a manual evaluation done by experts on a subset of the groups in each experiment. The PoS coherence is calculated automatically.

There is no universal widely accepted criteria for determining the semantic relations between two words. Two of the most common approaches are calculating WordNet distances and expert intuitions. We used both when evaluating the quality of the obtained groups.

For the WordNet similarity evaluation, we use the WordNet interface built in NLTK (Bird, Klein, and Loper, 2009). We calculate the Leacock-Chodorow Similarity[6] between each two words[7] in every group. We then sum all the obtained scores and divide them by the number of pairs to obtain average WordNet similarity for each method.

For the expert evaluation, we selected a subset of groups, generated in each experiment[8]. Three experts were asked to rate each group on a scale from 1 (unrelated) to 4 (strongly related)[9]. We calculate the average between all of the scores they gave on the groups of each experiment.

We define PoS coherence as the percent of words that belong to the most common PoS tag in each group. In order to calculate it, all obtained groups are automatically PoS tagged[10]. Then for each group, we count the percent of words that belong to each PoS and identify the most common tag.

4.3 Results

Table 4 shows the WordNet similarity evaluation. The average similarity score obtained by CLUTO is higher than the score obtained by Word2Vec (0.81-0.96 against 0.67-0.81). This indicates that the distances between the words in the CLUTO groups are shorter and the semantic relations are stronger. Increasing the corpus size improves the results for both CLUTO and Word2Vec. Preprocessing (specifically PoS tagging) improves the obtained results for all of the Word2Vec experiments. The groups obtained using Skip-Gram get lower scores in the evaluation compared with the groups obtained using CBOW.

Methodology	Corpus	Similarity
W2V-CBOW	4M (raw)	0.67
W2V-CBOW	4M (lemma)	0.67
W2V-CBOW	4M (pos)	0.72
W2V-CBOW	20M (raw)	0.74
W2V-CBOW	20M (lemma)	0.75
W2V-CBOW	20M (pos)	0.77
W2V-CBOW	40M (raw)	0.77
W2V-CBOW	40M (lemma)	0.78
W2V-CBOW	40M (pos)	0.81
W2V-SG	40M (raw)	0.69
W2V-SG	40M (lemma)	73
W2V-SG	40M (pos)	0.74
CLUTO	4M	0.81
CLUTO	20M	0.92
CLUTO	40M	0.96

Tabla 4: Wordnet Similarity

Table 5 shows the results from the expert evaluation of the semantic relations in the groups. The data is similar to the results with WordNet distances. The groups obtained by CLUTO show higher degree of semantic relatedness (2.8-3.4) compared to the groups obtained by Word2Vec (1.6-2.7). The CLUTO groups at 20M and 40M obtain average above 3, meaning that the experts consider all of the groups to be strongly related. For the experiments with Word2Vec, linguistic preprocessing improves the results, especially at biger corpus size (2.5 against 1.8 for 20M and 2.7 against 2 for 40M). The groups obtained using Skip-Gram algorithm are rated lower than the groups obtained using CBOW. The

[6] It calculates word similarity, based on the shortest path that connects the senses and the maximum depth of the taxonomy in which the senses occur.

[7] The calculation is based on the first sense of every word

[8] We selected the groups based on a word they contain - three verb groups (the ones that contain "say", "see", "want"), 3 noun groups ("person", "year", "hand"), 1 adjective group("good"), 1 adverb group("well). All of the selected words are among the 100 most commonly used words of English.)

[9] In the detailed description of the scale given to the experts: 1 corresponds to "no semantic relation"; 2 corresponds to "semantic relation between some words (less than 50 % of the group); 3 corresponds to "semantic relation between most of the words in the corpus (more than 50 %), but with multiple unrelated words"; 4 corresponds to "semantic relation between most of the words in the corpus, without many unrelated words"

[10] We use only the short PoS tag for this evaluation

preprocessed corpus obtains better groups, but the difference is smaller than the one observed with CBOW.

Methodology	Corpus	Score
W2V-CBOW	4M (raw)	1.6
W2V-CBOW	4M (lemma)	1.4
W2V-CBOW	4M (pos)	1.8
W2V-CBOW	20M (raw)	1.8
W2V-CBOW	20M (lemma)	2.4
W2V-CBOW	20M (pos)	2.5
W2V-CBOW	40M (raw)	2
W2V-CBOW	40M (lemma)	2.1
W2V-CBOW	40M (pos)	2.7
W2V-SG	40M (raw)	1.7
W2V-SG	40M (lemma)	1.8
W2V-SG	40M (pos)	2
CLUTO	4M	2.8
CLUTO	20M	3.2
CLUTO	40M	3.4

Tabla 5: Expert evaluation

Table 6 shows the results for the PoS coherence evaluation. The data shows that the groups obtained from CLUTO are more PoS coherent, compared with the groups obtained by Word2Vec (90-98 % against 69-81 %). For the corpora of size 20M and above, the groups obtained by CLUTO have almost 100 % PoS coherence, meaning that all of the lemmas belong to the same PoS. Both CLUTO and Word2Vec show improved results with the increase of corpus size. The results with Word2Vec indicate that corpus preprocessing largely improves the obtained results (69 %-73 % against 75 %-81 %). In fact, for this experiment the corpus preprocessing have bigger impact than the corpus size: a preprocessed corpus with a size of 4M generates more PoS coherent groups than raw 40M corpus (74-75 % against 73 %). The experiments with Skip-Gram obtain similar results for raw corpus. For Skip-Gram the preprocessed corpus also obtains better overall results, however lemmatized corpus obtains better results than the PoS tagged corpus.

Overall, all three evaluations identify similar patterns in the obtained clusters: (1) the groups obtained by CLUTO perform better than the groups obtained by Word2Vec; (2) Increasing the corpus size improves the quality of the results for both methodologies. This is true for semantic relatedness as well as for PoS coherence. The tendency to obtain more PoS coherent groups justifies the usage of PoS coherence as evaluation criteria; (3) Linguistic preprocessing improves the quality of the groups obtained by Word2Vec (with both algorithms).

Methodology	Corpus	PoS
W2V-CBOW	4M (raw)	69 %
W2V-CBOW	4M (lemma)	74 %
W2V-CBOW	4M (pos)	75 %
W2V-CBOW	20M (raw)	72 %
W2V-CBOW	20M (lemma)	77 %
W2V-CBOW	20M (pos)	80 %
W2V-CBOW	40M (raw)	73 %
W2V-CBOW	40M (lemma)	78 %
W2V-CBOW	40M (pos)	81 %
W2V-SG	40M (raw)	73 %
W2V-SG	40M (lemma)	80 %
W2V-SG	40M (pos)	77 %
CLUTO	4M	90 %
CLUTO	20M	97 %
CLUTO	40M	98 %

Tabla 6: PoS coherence

5 Conclusions and future work

This article compares two methodologies for identifying groups of semantically related words based on Distributional Semantic Models and vector representations. We applied the methodologies to a corpus of English and compared the quality of the obtained groups in terms of semantic relatedness and PoS coherence. We also analyzed the role of different factors, such as corpus size and linguistic preprocessing.

In the comparison of the two methodologies, the results show that CLUTO outperforms Word2Vec with respect to grouping, using corpora of medium size (20M - 40M). However, the quality of the results does depend on the size of the corpus. At 40M CLUTO already obtains very high quality results (98 % PoS coherence and 3.4/4 strength of semantic relationships in the evaluation of the experts) so further increase of the corpus is not likely to show large improvement. On the contrary at 40M Word2Vec still has room for improvement and we expect to narrow the difference between the two methodologies using much larger corpora (1B and above).

In the comparison of the different preprocessing corpora (i.e., raw, lemma, and PoS) in Word2Vec, the results show that lemmatization and PoS tagging largely improve the quality of the groups in both CBOW and Skip-Gram algorithms. This observation is consistent throughout all of the experiments and with respect to all of the evaluation criteria.

The presented comparison opens several lines of future research. First, the evaluation can be extended to bigger corpora, bigger number of vectors, and other languages. Second, the information provided and the suggested criteria for evaluation can be applied to other approaches to DSM and grouping. Finally, the different methodologies and preprocessing options can be evaluated in as part of more complex systems.

Acknowledgments

This work was supported by projects TIN2012-38603-C02-02, SGR-2014-623 and TIN2015-71147-C2-2.

We are grateful to Mariona Taulé, Horacio Rodríguez and the anonymous reviewers for their valuable comments.

References

Arppe, A., G. Gilquin, D. Glynn, M. Hilpert, and A. Zeschel. 2010. Cognitive corpus linguistics: five points of debate on current theory and methodology. *Corpora*, 5(1):1–27.

Baroni, M., S. Bernardini, A. Ferraresi, and E. Zanchetta. 2009. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Baroni, M. and A. Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Comput. Linguist.*, 36(4):673–721, December.

Bird, S., E. Klein, and E. Loper, 2009. *Natural Language Processing with Python.*

Boleda, G. and K. Erk. 2015. Distributional semantic features as semantic primitives – or not.

Erk, K. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.

Firth, J. R. 1957. A synopsis of linguistic theory 1930-55. 1952-59:1–32.

Harris, Z. 1954. Distributional structure. *Word*, 10(23):146–162.

Karypis, G. 2002. CLUTO a clustering toolkit. Technical Report 02-017, Dept. of Computer Science, University of Minnesota.

Lenci, A. 2008. Distributional semantics in linguistic and cognitive research. *Rivista di Linguistica*, 20(1):1–31.

Martí, M. A., M. Taulé, V. Kovatchev, and M. Salamó. Submitted, 2016. Discover: Distributional approach based on syntactic dependencies for discovering constructions. *Natural Language Engineering.*

Mikolov, T., K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Moisl, H. 2015. *Cluster Analisys for Corpus Linguistics.* De Gruyter Mouton.

Turney, P. D. and P. Pantel. 2010. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January.