JoBimText Visualizer: A Graph-based Approach to Contextualizing Distributional Similarity

Alfio Gliozzo[1] Chris Biemann[2] Martin Riedl[2]
Bonaventura Coppola[1] Michael R. Glass[1] Matthew Hatem[1]
(1) IBM T.J. Watson Research, Yorktown Heights, NY 10598, USA
(2) FG Language Technology, CS Dept., TU Darmstadt, 64289 Darmstadt, Germany
{gliozzo,mrglass,mhatem}@us.ibm.com coppolab@gmail.com
{biem,riedl}@cs.tu-darmstadt.de

Abstract

We introduce an interactive visualization component for the JoBimText project. JoBimText is an open source platform for large-scale distributional semantics based on graph representations. First we describe the underlying technology for computing a distributional thesaurus on words using bipartite graphs of words and context features, and contextualizing the list of semantically similar words towards a given sentential context using graph-based ranking. Then we demonstrate the capabilities of this contextualized text expansion technology in an interactive visualization. The visualization can be used as a semantic parser providing contextualized expansions of words in text as well as disambiguation to word senses induced by graph clustering, and is provided as an open source tool.

1 Introduction

The aim of the JoBimText[1] project is to build a graph-based unsupervised framework for computational semantics, addressing problems like lexical ambiguity and variability, word sense disambiguation and lexical substitutability, paraphrasing, frame induction and parsing, and textual entailment. We construct a semantic analyzer able to self-adapt to new domains and languages by unsupervised learning of semantics from large corpora of raw text. At the moment, this analyzer encompasses contextualized similarity, sense clustering, and a mapping of senses to existing knowledge bases. While its primary target application is functional domain adaptation of Question Answering (QA) systems (Ferrucci et al., 2013), output of the semantic analyzer has been successfully utilized for word sense disambiguation (Miller et al., 2012) and lexical substitution (Szarvas et al., 2013). Rather than presenting the different algorithms and technical solutions currently implemented by the JoBimText community in detail, in this paper we will focus on available functionalities and illustrate them using an interactive visualization.

[1] http://sf.net/projects/jobimtext/

2 Underlying Technologies

While distributional semantics (de Saussure, 1959; Harris, 1951; Miller and Charles, 1991) and the computation of distributional thesauri (Lin, 1998) has been around for decades, its full potential has yet to be utilized in Natural Language Processing (NLP) tasks and applications. Structural semantics claims that meaning can be fully defined by semantic oppositions and relations between words. In order to perform a reliable knowledge acquisition process in this framework, we gather statistical information about word co-occurrences with syntactic contexts from very large corpora. To avoid the intrinsic quadratic complexity of the similarity computation, we have developed an optimized process based on MapReduce (Dean and Ghemawat, 2004) that takes advantage of the sparsity of contexts, which allows scaling the process through parallelization. The result of this computation is a graph connecting the most discriminative contexts to terms and explicitly linking the most similar terms. This graph represents local models of semantic relations *per term* rather than a model with fixed dimensions. This representation departs from the vector space metaphor (Schütze, 1993; Erk and Padó, 2008; Baroni and Zamparelli,

Proceedings of the TextGraphs-8 Workshop, pages 6–10,
Seattle, Washington, USA, 18 October 2013.

2010), commonly employed in other frameworks for distributional semantics such as LSA (Deerwester et al., 1990) or LDA (Blei et al., 2003).

The main contribution of this paper is to describe how we operationalize semantic similarity in a graph-based framework and explore this semantic graph using an interactive visualization. We describe a scalable and flexible computation of a distributional thesaurus (DT), and the contextualization of distributional similarity for specific occurrences of language elements (i.e. terms). For related works on the computation of distributional similarity, see e.g. (Lin, 1998; Lin and Dyer, 2010).

2.1 Holing System

To keep the framework flexible and abstract with respect to the pre-processing that identifies structure in language material, we introduce the holing operation, cf. (Biemann and Riedl, 2013). It is applied to observations over the structure of text, and splits these observations into a pair of two parts, which we call the "Jo" and the "Bim"[2]. All JoBim pairs are maintained in the bipartite First-Order JoBim graph $TC(T, C, E)$ with T set of terms (Jos), C set of contexts (Bims), and $e(t, c, f) \in E$ edges between $t \in T$, $c \in C$ with frequency f. While these parts can be thought of as language elements referred to as *terms*, and their respective *context features*, splits over arbitrary structures are possible (including pairs of terms for Jos), which makes this formulation more general than similar formulations found e.g. in (Lin, 1998; Baroni and Lenci, 2010). These splits form the basis for the computation of global similarities and for their contextualization. A Holing System based on dependency parses is illustrated in Figure 1: for each dependency relation, two JoBim pairs are generated.

2.2 Distributed Distributional Thesaurus Computation

We employ the Apache Hadoop MapReduce Framwork[3], and Apache Pig[4], for parallelizing and distributing the computation of the DT. We describe this computation in terms of graph transformations.

[2]arbitrary names to emphasize the generality, should be thought of as "term" and "context"

[3]`http://hadoop.apache.org`

[4]`http://pig.apache.org/`

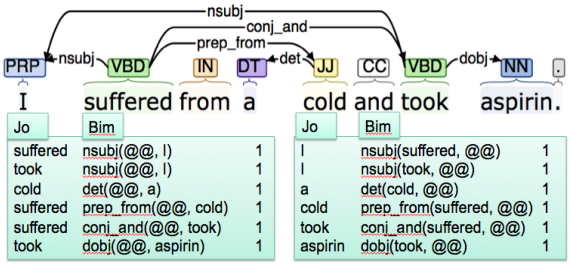

Figure 1: Jos and Bims generated applying a dependency parser (de Marneffe et al., 2006) to the sentence *I suffered from a cold and took aspirin*. The @@ symbolizes the hole.

Staring from the JoBim graph TC with counts as weights, we first apply a statistical test[5] to compute the significance of each pair (t, c), then we only keep the p most significant pairs per t. This constitutes our first-order graph for Jos FO_{JO}. In analogy, when keeping the p most significant pairs per c, we can produce the first-order graph for Bims FO_{BIM}. The second order similarity graph for Jos is defined as $SO_{JO}(T, E)$ with Jos $t_1, t_2 \in T$ and undirected edges $e(t_1, t_2, s)$ with similarity $s = |\{c|e(t_1, c) \in FO_{JO}, e(t_2, c) \in FO_{JO}\}|$, which defines similarity between Jos as the number of salient features two Jos share. SO_{JO} defines a distributional thesaurus. In analogy, SO_{BIM} is defined over the shared Jos for pairs of Bims and defines similarity of contexts. This method, which can be computed very efficiently in a few MapReduce steps, has been found superior to other measures for very large datasets in semantic relatedness evaluations in (Biemann and Riedl, 2013), but could be replaced by any other measure without interfering with the remainder of the system.

2.3 Contextualization with CRF

While the distributional thesaurus provides the similarity between pairs of terms, the fidelity of a particular expansion depends on the context. From the term-context associations gathered in the construction of the distributional thesaurus we effectively have a language model, factorized according to the holing operation. As with any language model, smoothing is critical to performance. There may be

[5]we use log-likelihood ratio (Dunning, 1993) or LMI (Evert, 2004)

many JoBim (term-context) pairs that are valid and yet under represented in the corpus. Yet, there may be some similar term-context pair that is attested in the corpus. We can find similar contexts by expanding the term arguments with similar terms. However, again we are confronted with the fact that the similarity of these terms depends on the context.

This suggests some technique of joint inference to expand terms in context. We use marginal inference in a conditional random field (CRF) (Lafferty et al., 2001). A particular world, $\mathbf{x}$ is defined as single, definite sequence of either original or expanded words. The weight of the world, $w(\mathbf{x})$ depends on the degree to which the term-context associations present in this sentence are present in the corpus and the general out-of-context similarity of each expanded term to the corresponding term in the original sentence. Therefore the probability associated with any expansion t for any position x_i is given by Equation 1. Where Z is the partition function, a normalization constant.

$$P(x_i = t) = \frac{1}{Z} \sum_{\{\mathbf{x} \mid x_i = t\}} e^{w(\mathbf{x})} \quad (1)$$

The balance between the plausibility of an expanded sentence according to the language model, and its per-term similarity to the original sentence is an application specific tuning parameter.

2.4 Word Sense Induction, Disambiguation and Cluster Labeling

The contextualization described in the previous subsection performs implicit word sense disambiguation (WSD) by ranking contextually better fitting similar terms higher. To model this more explicitly, and to give rise to linking senses to taxonomies and domain ontologies, we apply a word sense induction (WSI) technique and use information extracted by IS-A-patterns (Hearst, 1992) to label the clusters.

Using the aggregated context features of the clusters, the word cluster senses are assigned in context. The DT entry for each term j as given in $SO_{JO}(J, E)$ induces an open neighborhood graph $N_j(V_j, E_j)$ with $V_j = \{j' | e(j, j', s) \in E)$ and E_j the projection of E regarding V_j , consisting of similar terms to j and their similarities, cf. (Widdows and Dorow, 2002).

We cluster this graph using the Chinese Whispers graph clustering algorithm (Biemann, 2010), which finds the number of clusters automatically, to obtain induced word senses. Running shallow, part-of-speech-based IS-A patterns (Hearst, 1992) over the text collection, we obtain a list of extracted IS-A relationships between terms, and their frequency. For each of the word clusters, consisting of similar terms for the same target term sense, we aggregate the IS-A information by summing the frequency of hypernyms, and multiplying this sum by the number of words in the cluster that elicited this hypernym. This results in taxonomic information for labeling the clusters, which provides an abstraction layer for terms in context[6]. Table 1 shows an example of this labeling from the model described below. The most similar 200 terms for "jaguar" have been clustered into the car sense and the cat sense and the highest scoring 6 hypernyms provide a concise description of these senses. This information can be used to automatically map these cluster senses to senses in an taxonomy or ontology. Occurrences of ambiguous words in context can be disambiguated to these cluster senses comparing the actual context with salient contexts per sense, obtained by aggregating the Bims from the FO_{JO} graph per cluster.

sense	IS-A labels	similar terms
jaguar N.0	car, brand, company, automaker, manufacturer, vehicle	geely, lincoln-mercury, tesla, peugeot, ..., mitsubishi, cadillac, jag, benz, mclaren, skoda, infiniti, sable, thunderbird
jaguar N.1	animal, species, wildlife, team, wild animal, cat	panther, cougar, alligator, tiger, elephant, bull, hippo, dragon, leopard, shark, bear, otter, lynx, lion

Table 1: Word sense induction and cluster labeling example for "jaguar". The shortened cluster for the car sense has 186 members.

3 Interactive Visualization

3.1 Open Domain Model

The open domain model used in the current visualization has been trained from newspaper cor-

[6]Note that this mechanism also elicits hypernyms for unambiguous terms receiving a single cluster by the WSI technique.

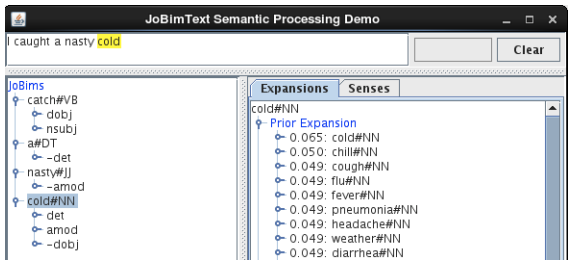

Figure 2: Visualization GUI with prior expansions for "cold". Jobims are visualized on the left, expansions on the right side.

pora using 120 million sentences (about 2 Gigawords), compiled from LCC (Richter et al., 2006) and the Gigaword (Parker et al., 2011) corpus. We constructed a UIMA (Ferrucci and Lally, 2004) pipeline, which tokenizes, lemmatizes and parses the data using the Stanford dependency parser (de Marneffe et al., 2006). The last annotator in the pipeline annotates *Jo*s and *Bim*s using the collapsed dependency relations, cf. Fig. 1. We define the lemmatized forms of the terminals including the part-of-speech as Jo and the lemmatized dependent word and the dependency relation name as Bim.

3.2 Interactive Visualization Features

Evaluating the impact of this technology in applications is an ongoing effort. However, in the context of this paper, we will show a visualization of the capabilities allowed by this flavor of distributional semantics. The visualization is a GUI as depicted in Figure 2, and exemplifies a set of capabilities that can be accessed through an API. It is straightforward to include all shown data as features for semantic preprocessing. The input is a sentence in natural language, which is processed into JoBim pairs as described above. All the Jos can be expanded, showing their paradigmatic relations with other words.

We can perform this operation with and without taking the context into account (cf. Sect. 2.3). The latter performs an implicit disambiguation by ranking similar words higher if they fit the context. In the example, the "common cold" sense clearly dominates in the prior expansions. However, "weather" and "chill" appear amongst the top-similar prior expansions.

We also have implemented a sense view, which displays sense clusters for the selected word, see Figure 3. Per sense, a list of expansions is provided together with a list of possible IS-A types. In this example, the algorithm identified two senses of "cold" as a temperature and a disease (not all cluster members shown). Given the JoBim graph of the context (as displayed left in Fig. 2), the particular occurrence of "cold" can be disambiguated to Cluster 0 in Fig. 3, since its Bims "amod(@@,nasty)" and "-dobj(catch, @@)" are found in FO_{JO} for far more members of cluster 0 than for members of cluster 1. Applications of this type of information include knowledge-based word sense disambiguation (Miller et al., 2012), type coercion (Kalyanpur et al., 2011) and answer justification in question answering (Chu-Carroll et al., 2012).

4 Conclusion

In this paper we discussed applications of the JoBimText platform and introduced a new interactive visualization which showcases a graph-based unsupervised technology for semantic processing. The implementation is operationalized in a way that it can be efficiently trained "off line" using MapReduce, generating domain and language specific models for distributional semantics. In its "on line" use, those models are used to enhance parsing with contextualized text expansions of terms. This expansion step is very efficient and runs on a standard laptop, so it can be used as a semantic text preprocessor. The entire project, including pre-computed data models, is available in open source under the ASL 2.0, and allows computing contextualized lexical expansion on arbitrary domains.

References

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

M. Baroni and R. Zamparelli. 2010. Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space. In *Proc. EMNLP-2010*, pages 1183–1193, Cambridge, Massachusetts.

C. Biemann and M. Riedl. 2013. Text: Now in 2D! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95.

C. Biemann. 2010. Co-occurrence cluster features for lexical substitutions in context. In *Proceedings of TextGraphs-5*, pages 55–59, Uppsala, Sweden.

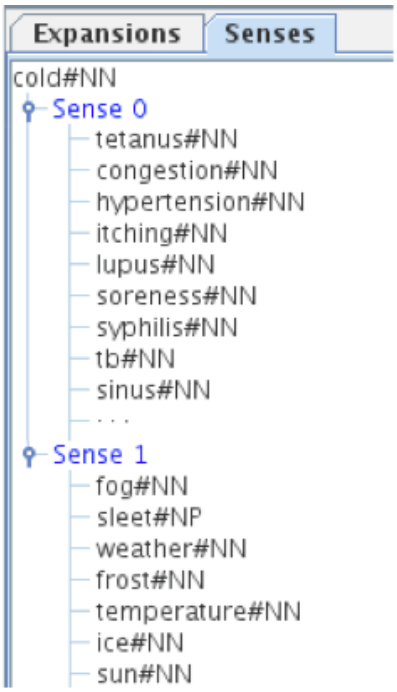

Figure 3: Senses induced for the term "cold".

D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March.

J. Chu-Carroll, J. Fan, B. K. Boguraev, D. Carmel, D. Sheinwald, and C. Welty. 2012. Finding needles in the haystack: search and candidate generation. *IBM J. Res. Dev.*, 56(3):300–311.

M.-C. de Marneffe, B. MacCartney, and C. D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. LREC-2006*, Genova, Italy.

Ferdinand de Saussure. 1916. *Cours de linguistique générale*. Payot, Paris, France.

J. Dean and S. Ghemawat. 2004. MapReduce: Simplified Data Processing on Large Clusters. In *Proc. OSDI '04*, San Francisco, CA.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

T. Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

K. Erk and S. Padó. 2008. A structured vector space model for word meaning in context. In *Proc. EMNLP-2008*, pages 897–906, Honolulu, Hawaii.

S. Evert. 2004. *The statistics of word cooccurrences: word pairs and collocations*. Ph.D. thesis, IMS, Universität Stuttgart.

D. Ferrucci and A. Lally. 2004. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. In *Nat. Lang. Eng. 2004*, pages 327–348.

D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, and E. T. Mueller. 2013. Watson: Beyond Jeopardy! *Artificial Intelligence*, 199-200:93–105.

Z. S. Harris. 1951. *Methods in Structural Linguistics*. University of Chicago Press, Chicago.

M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. COLING-1992*, pages 539–545, Nantes, France.

A. Kalyanpur, J.W. Murdock, J. Fan, and C. Welty. 2011. Leveraging community-built knowledge for type coercion in question answering. In *Proc. ISWC 2011*, pages 144–156. Springer.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML 2001*, pages 282–289, San Francisco, CA, USA.

J. Lin and C. Dyer. 2010. *Data-Intensive Text Processing with MapReduce*. Morgan & Claypool Publishers, San Rafael, CA.

D. Lin. 1998. Automatic retrieval and clustering of similar words. In *Proc. COLING-98*, pages 768–774, Montréal, Quebec, Canada.

G. A. Miller and W. G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

T. Miller, C. Biemann, T. Zesch, and I. Gurevych. 2012. Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. In *Proc. COLING-2012*, pages 1781–1796, Mumbai, India.

R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda. 2011. *English Gigaword Fifth Edition*. Linguistic Data Consortium, Philadelphia.

M. Richter, U. Quasthoff, E. Hallsteinsdóttir, and C. Biemann. 2006. Exploiting the leipzig corpora collection. In *Proc. IS-LTC 2006*, Ljubljana, Slovenia.

H. Schütze. 1993. Word space. In *Advances in Neural Information Processing Systems 5*, pages 895–902. Morgan Kaufmann.

G. Szarvas, C. Biemann, and I. Gurevych. 2013. Supervised all-words lexical substitution using delexicalized features. In *Proc. NAACL-2013*, Atlanta, GA, USA.

D. Widdows and B. Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proc. COLING-2002*, pages 1–7, Taipei, Taiwan.