Unsupervised Natural Language Processing using Graph Models

Chris Biemann

NLP Dept., University of Leipzig

Johannisgasse 26

04103 Leipzig, Germany

biem@informatik.uni-leipzig.de

Abstract

In the past, NLP has always been based on the explicit or implicit use of linguistic knowledge. In classical computer linguistic applications *explicit* rule based approaches prevail, while machine learning algorithms use *implicit* knowledge for generating linguistic knowledge. The question behind this work is: how far can we go in NLP without assuming explicit or implicit linguistic knowledge? How much efforts in annotation and resource building are needed for what level of sophistication in text processing? This work tries to answer the question by experimenting with algorithms that do *not presume any* linguistic knowledge in the system. The claim is that the knowledge needed can largely be acquired by knowledge-free and unsupervised methods. Here, graph models are employed for representing language data. A new graph clustering method finds related lexical units, which form word sets on various levels of homogeneity. This is exemplified and evaluated on language separation and unsupervised part-of-speech tagging, further applications are discussed.

1 Introduction

1.1 Unsupervised and Knowledge-Free

A frequent remark on work dealing with unsupervised methods in NLP is the question: "Why not take linguistic knowledge into account?" While for English, annotated corpora, classification examples, sets of rules and lexical semantic word nets of high coverage do exist, this does not reflect the situation for most of even the major world languages. Further, as e.g. Lin (1997) notes, handmade and generic resources often do not fit the application domain, whereas resources created from and for the target data will not suffer from these discrepancies.

Shifting the workload from creating resources manually to developing generic methods, a one-size-fits-all solution needing only minimal adaptation to new domains and other languages comes into reach.

1.2 Graph Models

The interest in incorporating graph models into NLP arose quite recently, and there is still a high potential exploiting this combination (cf. Widdows, 2005). An important parallelism between human language and network models is the small world structure of lexical networks both built manually and automatically (Steyvers and Tenenbaum, 2005), providing explanation for power-law distributions like Zipf's law and others, see Biemann (2007). For many problems in NLP, a graph representation is an intuitive, natural and direct way to represent the data.

The pure vector space model (cf. Schütze, 1993) is not suited to highly skewed distributions omni-present in natural language. Computationally expensive, sometimes lossy transformations have to be applied for effectiveness and efficiency in processing. Graph models are a veritable alternative, as the equivalent of zero-entries in the vector representation are neither represented nor have to

be processed, rendering dimensionality reduction techniques unnecessary while still retaining the exact information.

1.3 Roadmap

For the entirety of this research, nothing more is required as input data than plain, tokenized text, separated into sentences. This is surely quite a bit of knowledge that is provided to the system, but unsupervised word boundary and sentence boundary detection is left for future work. Three steps are undertaken to identify similar words on different levels of homogeneity: same language, same part-of-speech, or same distributional properties. Figure 1 shows a coarse overview of the processing steps discussed in this work.

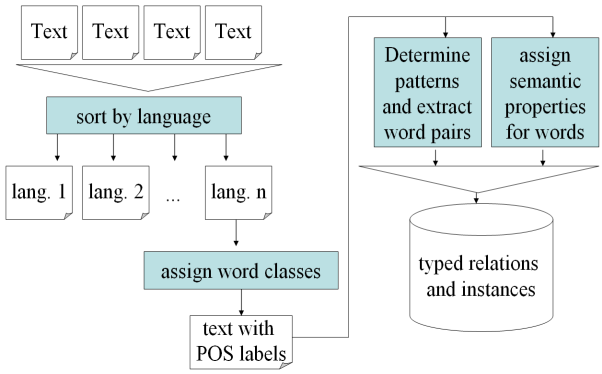

Figure 1: Coarse overview: From multilingual input to typed relations and instances

2 Methods in Unsupervised Processing

Having at hand neither explicit nor implicit knowledge, but in turn the goal of identifying structure of equivalent function, the only possibility that is left in unsupervised and knowledge-free processing is statistics and clustering.

2.1 Co-occurrence Statistics

As a building block, co-occurrence statistics are used in several components of the system described here. A significance measure for co-occurrence is a means to distinguish between observations that are there by chance and effects that take place due to an underlying structure. Throughout, the likelihood ratio (Dunning, 1993) is used as significance measure because of its stable performance in various evaluations, yet many more measures are possible. Dependent on the context range in co-occurrence calculation, they will be called sentence-based or neighbor-based co-occurrences in the remainder of this paper. The entirety of all co-occurrences of a corpus is called its co-occurrence graph. Edges are weighted by co-occurrence significance; often a threshold on edge weight is applied.

2.2 Graph Clustering

For clustering graphs, a plethora of algorithms exist that are motivated from a graph-theoretic viewpoint, but often optimize NP-complete measures (cf. Šíma and Schaeffer, 2005), making them non-applicable to lexical data that is naturally represented in graphs with millions of vertices. In Biemann and Teresniak (2005) and more detailed in Biemann (2006a), the Chinese Whispers (CW) Graph Clustering algorithm is described, which is a randomized algorithm with edge-linear run-time. The core idea is that vertices retain class labels which are inherited along the edges: In an update step, a vertex gets assigned the predominant label in its neighborhood. For initialization, all vertices get different labels, and after a handful of update steps per vertex, almost no changes in the labeling are observed – especially small world graphs converge fast. CW can be viewed as a more efficient modification and simplification of Markov Chain Clustering (van Dongen, 2000), which requires full matrix multiplications.

CW is parameter-free, non-deterministic and finds the number of clusters automatically – a feature that is welcome in NLP, where the number of desired clusters (e.g. in word sense induction) is often unknown.

3 Results

3.1 Language Separation

Clustering the sentence-based co-occurrence graph of a multilingual corpus with CW, a language separator with almost perfect performance is implemented in the following way: The clusters represent languages; a sentence gets assigned the label of the cluster with the highest lexical overlap between sentence and cluster. The method is evaluated in (Biemann and Teresniak, 2005) by sorting monolingual material that has been artificially mixed together. Dependent on similarities of languages, the method works almost error-free from about 100-1,000 sentences per language on. For

languages with different encoding, it is possible to un-mix corpora of size factors up to 10,000 for the monolingual parts.

In a nutshell, comparable scores to supervised language identifiers are reached without training. Notice that the number of languages in a multilingual chunk of text is unknown. This prohibits any clustering method that needs the number of clusters to be specified be-forehand.

3.2 Unsupervised POS Tagging

Unlike in standard POS tagging, there is neither a set of predefined categories, nor annotation in a text. As POS tagging is not a system for its own sake, but serves as a preprocessing step for systems building upon it, the names and the number of categories are very often not important.

The system presented in Biemann (2006b) uses CW clustering on graphs constructed by distributional similarity to induce a lexicon of supposedly non-ambiguous words w.r.t. POS by selecting only safe bets and excluding questionable cases from the lexicon. In this implementation, two clusterings are combined, one for high and medium frequency words, the other collecting medium and low frequency words. High and medium frequency words are clustered by similarity of their stop word context feature vectors: a graph is built, including only words that are involved in highly similar pairs. Clustering this graph of typically 5,000 vertices results in several hundred clusters, which are further used as POS categories. To extend the lexicon, words of medium and low frequency are clustered using a graph that encodes similarity of neighbor-based co-occurrences. Both clusterings are mapped by overlapping elements into a lexicon that provides POS information for some 50,000 words. For obtaining a clustering on datasets of this size, an effective algorithm like CW is crucial. Using this lexicon, a trigram tagger with a morphological extension is trained, which assigns a tag to every token in the corpus.

The tagsets obtained with this method are usually more fine-grained than standard tagsets and reflect syntactic as well as semantic similarity. Figure 2 demonstrates the domain-dependence on the tagset for MEDLINE: distinguishing e.g. illnesses and error probabilities already in the tagset might be a valuable feature for relation extraction tasks.

Size	Sample words
1613	colds, apnea, aspergilloma, ACS, breathlessness, lesions, perforations, ...
1383	proven, supplied, engineered, distinguished, constrained, omitted, …
589	dually, circumferentially, chronically, rarely, spectrally, satisfactorily, ...
124	1-min, two-week, 4-min, 2-day, …
6	P<0.001, P<0.01, p<0.001, p<0.01, ...

Figure 2: Some examples for MEDLINE tagset: Number of lex. entries per tag and sample words.

In Biemann (2006b), the tagger output was directly compared to supervised taggers for English, German and Finnish via information-theoretic measures. While it is possible to compare the contribution of different components of a system relatively along this scale, it only gives a poor impression on the utility of the unsupervised tagger's output. Therefore, the tagger was evaluated indirectly in machine learning tasks, where POS tags are used as features. Biemann et al. (2007) report that for standard Named Entity Recognition, Word Sense Disambiguation and Chunking tasks, using unsupervised POS tags as features helps about as much as supervised tagging: Overall, almost no significant differences between results could be observed, supporting the initial claim.

3.3 Word Sense Induction (WSI)

Co-occurrences are a widely used data source for WSI. The methodology of Dorow and Widdows (2003) was adopted: for the focus word, obtain its graph neighborhood (all vertices that are connected via edges to the focus word vertex and edges between these). Clustering this graph with CW and regarding clusters as senses, this method yields comparable results to Bordag (2006), tested using the unsupervised evaluation framework presented there. More detailed results are reported in Biemann (2006a).

4 Further Work

4.1 Word Sense Disambiguation (WSD)

The encouraging results in WSI enable support in automatic WSD systems. As described by Agirre et al. (2006), better performance can be expected if the WSI component distinguishes between a large number of so-called micro-senses. This illustrates a

principle of unsupervised NLP: It is not important to reproduce word senses found by introspection; rather, it is important that different usages of a word can be reliably distinguished, even if the corresponding WordNet sense is split into several subsenses.

4.2 Distributional Thesaurus with Relations

It is well understood that distributional similarity reflects semantic similarity and can be used to automatically construct a distributional thesaurus for frequent words (Lin, 1997; inter al). Until now, most works aiming at semantic similarity rely on a parser that extracts dependency relations. The claim here again is that similarity on parser output might be replaced by similarity on a pattern basis, (cf. Davidov and Rappoport 2006). For class-based generalization in these patterns, the system described in section 3.2 might prove useful. Preliminary experiments revealed that similarity on significantly co-occurring patterns is able to produce very promising similarity rankings. A clustering of these with CW leads to thesaurus entries comparable to thesauri like Roget's.

Clustering not only words based on similarity of patterns, but also patterns based on similarity of words enables us to identify clusters of patterns with different relations they manifest.

5 Conclusion

The claim of this work is that unsupervised NLP can support and/or replace preprocessing steps in NLP that have previously been achieved by a large amount of manual work, i.e. annotation, rule construction or resource building. This is proven empirically on the tasks of language identification and part-of-speech tagging, exemplified on WSD and discussed for thesaurus construction and relation extraction. The main contributions of the dissertation that is summarized here are:

- A framework for unsupervised NLP
- An efficient graph clustering algorithm
- An unsupervised language separator
- An unsupervised POS tagger

The main advantage of unsupervised NLP, namely language independence, will enable the immediate processing of all languages and domains for which a large amount of text is electronically available.

References

E. Agirre, D. Martínez, O. López de Lacalle and A. Soroa. 2006. *Evaluating and optimizing the parameters of an unsupervised graph-based WSD algorithm*. Proceedings of Textgraphs-06, New York, USA

C. Biemann and S. Teresniak. 2005. *Disentangling from Babylonian Confusion – Unsupervised Language Identification*. Proc. CICLing-2005, Mexico City

C. Biemann. 2006a. *Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems*. Proceedings of Textgraphs-06, New York, USA

C. Biemann. 2006b. *Unsupervised Part-of-Speech Tagging Employing Efficient Graph Clustering*. Proceedings of COLING/ACL-06 SRW, Sydney, Australia

C. Biemann. 2007. *A Random Text Model for the Generation of Statistical Language Invariants*. Proceedings of HLT-NAACL-07, Rochester, USA

C. Biemann, C. Giuliano and A. Gliozzo. 2007. *Unsupervised POS tagging supporting supervised methods*. Manuscript to appear

S. Bordag. 2006. *Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation*. Proceedings of EACL-06. Trento, Italy

D. Davidov, A. Rappoport. 2006. *Efficient Unsupervised Discovery of Word Categories Using Symmetric Patterns and High Frequency Words*. Proceedings of COLING/ACL-06, Sydney, Australia

S. van Dongen. 2000. *A cluster algorithm for graphs*. Technical Report INS-R0010, CWI, Amsterdam.

B. Dorow and D. Widdows. 2003. *Discovering Corpus-Specific Word Senses*. In EACL-2003 Conference Companion, Budapest, Hungary

T. Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1), pp. 61-74

D. Lin. 1997. *Automatic Retrieval and Clustering of Similar Words*. Proc. COLING-97, Montreal, Canada

H. Schütze. 1993. *Word Space*. Proceedings of NIPS-5, Morgan Kaufmann, San Francisco, CA, USA

J. Šíma and S.E. Schaeffer. 2005. *On the NP-completeness of some graph cluster measures*. Technical Report cs.CC/0506100, http://arxiv.org/.

M. Steyvers, J. B. Tenenbaum. 2005. The large-scale structure of semantic networks. *Cog. Science*, 29(1)

D. Widdows. 2005. *Geometry and Meaning*. CSLI Lecture notes #172, Stanford, USA