A Quantitative Evaluation of Global Word Sense Induction

Marianna Apidianaki and Tim Van de Cruys

Alpage, INRIA & University Paris 7
175 rue du Chevaleret
75013, Paris, France
Marianna.Apidianaki@inria.fr, Tim.Van_de_Cruys@inria.fr

Abstract. Word sense induction (WSI) is the task aimed at automatically identifying the senses of words in texts, without the need for handcrafted resources or annotated data. Up till now, most WSI algorithms extract the different senses of a word 'locally' on a per-word basis, i.e. the different senses for each word are determined separately. In this paper, we compare the performance of such algorithms to an algorithm that uses a 'global' approach, i.e. the different senses of a particular word are determined by comparing them to, and demarcating them from, the senses of other words in a full-blown word space model. We adopt the evaluation framework proposed in the SemEval-2010 Word Sense Induction & Disambiguation task. All systems that participated in this task use a local scheme for determining the different senses of a word. We compare their results to the ones obtained by the global approach, and discuss the advantages and weaknesses of both approaches.

1 Introduction

Word sense induction (WSI) methods automatically identify the senses of words in texts, without the need for predefined resources or annotated data. These methods offer an alternative to the use of expensive hand-crafted resources developed according to the 'fixed list of senses' paradigm, which present several drawbacks for efficient semantic processing [1]. The assumption underlying unsupervised WSI methods is the distributional hypothesis of meaning [2], according to which words that occur in similar contexts tend to be similar. In distributional semantic analysis, the co-occurrences of words in texts constitute the features that serve to calculate their similarity. Following this approach, data-driven WSI algorithms calculate the similarity of the contexts of polysemous target words and group them into clusters. The resulting clusters describe the target word senses.

The unsupervised algorithms used for WSI can be distinguished into *local* and *global*. Local algorithms work on a per-word basis, determining the senses for each word separately. Algorithms that use a global approach determine the different senses of a particular word by comparing them to, and demarcating them from, the senses of other words in a full-blown word space model.

In this paper, we compare the performance of these two types of algorithms for sense induction. The comparison is carried out using the evaluation framework proposed in the SemEval-2010 Word Sense Induction & Disambiguation (WSI&D) task [3, 4]. The SemEval WSI tasks [4, 5] provide a common ground for comparison and evaluation of different sense induction and discrimination systems. All the systems that participated in the SemEval-2010 WSI&D task use a local scheme for determining the different senses of a word. We compare their results to the ones obtained by the global approach, and discuss the advantages and weaknesses of both approaches.

The paper is organized as follows. We first explain how word senses are identified in the local and the global approaches to sense induction, and we present the global algorithm used in our research. Section 3 describes the evaluation setting that we adopt and the metrics that will be used in order to evaluate the performance of the algorithms. In Section 4, we present the evaluation results of the global approach, and compare them to the results obtained by the local systems that participated in the SemEval-2010 WSI&D task. Our last section draws conclusions, and lays out some avenues for future work.

2 WSI algorithms

2.1 Inducing word senses on a per-word basis

Local methods to word sense induction discover the senses of a target word (w) by clustering its instances in texts according to their semantic similarity. Following the distributional hypothesis of meaning, words that are used in similar contexts carry similar meanings [2, 6]. So, the instances of w that appear in similar contexts are considered as semantically similar and its senses can be discovered by clustering its contexts [7].

The features used for calculating the similarity of the instances of w are their co-occurrences in a fixed-sized window of text. So, the different instances of w in a corpus can be represented by feature vectors created from their contexts [8, 9]. The grouping of the context vectors according to their similarity generates a number of clusters that describe the different senses of w. The context of w may be taken into account in different ways : it may be modeled as a first-order context vector, representing the direct context of the instances of w in the corpus [10, 11], or by using higher-order vectors, i.e. by considering the context vectors of the words occurring in the target context [12].

Other methods use the words found in the context of target words in order to construct co-occurrence graphs. In a graph of this type, the vertices correspond to the words appearing in the contexts of the target words and the edges represent their relations. These relations may be grammatical [13] or they may be co-occurrences of the words in fixed contexts [14, 15]. The senses of the target words are discovered by partitioning the co-occurrence graph using clustering techniques, or by using a PageRank algorithm.

2.2 Global approach to sense induction

In contrast to the local approach to sense induction, where senses are discovered by clustering contexts for each word individually, the global approach discovers senses by clustering semantically similar senses of words in a global manner, comparing them and demarcating them from the senses of other words in a full-blown word space model. The similarity between the senses is calculated on the basis of their common features, e.g. the syntactic dependencies a particular sense occurs with [16]. In Pantel and Lin's [17] method, the similarity of word senses is calculated on the basis of the dependency relations in which the senses take part (extracted from a syntactically annotated corpus). Each word is represented by a feature vector, where each feature corresponds to a syntactic context (dependency triple) in which the word occurs. Each feature is weighted and its value corresponds to the pointwise mutual information between the feature and the word. The algorithm first discovers a set of tight clusters (called 'committees') in the similarity space. Each word is then assigned to the closest committee by comparing the word's feature vector to the centroid of a committee (i.e. the mean of the feature vectors of the committee members). After a word is assigned to a particular committee, the overlapping features are deleted from the word's vector, which allows for the discovery of less dominant senses. Each cluster that a word belongs to describes one of its senses.

2.3 Non-negative Matrix Factorization for sense induction

Sense induction The global algorithm implemented here is based on the one proposed by Van de Cruys [18]. This algorithm creates semantic word models by using an extension of non-negative matrix factorization (NMF) [19], that combines both the bag of words approach and the syntax-based approach to sense induction. The intuition in this is that the syntactic features of the syntax-based approach can be disambiguated by the semantic dimensions found by the bag of words approach. The algorithm finds a small number of latent semantic dimensions, according to which nouns, contexts and syntactic relations are classified.

Nouns are classified according to both bag-of-words context and syntactic context, so three matrices are constructed that capture the co-occurrence frequency information for each mode. The first matrix contains co-occurrence frequencies of nouns cross-classified by dependency relations, the second matrix contains co-occurrence frequencies of nouns cross-classified by words that appear in the noun's context window, and the third matrix contains co-occurrence frequencies of dependency relations cross-classified by co-occurring context words. NMF is then applied to the three matrices and the separate factorizations are interleaved (i.e. the results of the former factorization are used to initialize the factorization of the next matrix). A graphical representation of the interleaved factorization algorithm is given in figure 1.

When the factorization is finished, the three different modes (nouns, bag-of-words context words and syntactic relations) are all represented as a limited number of semantic dimensions.

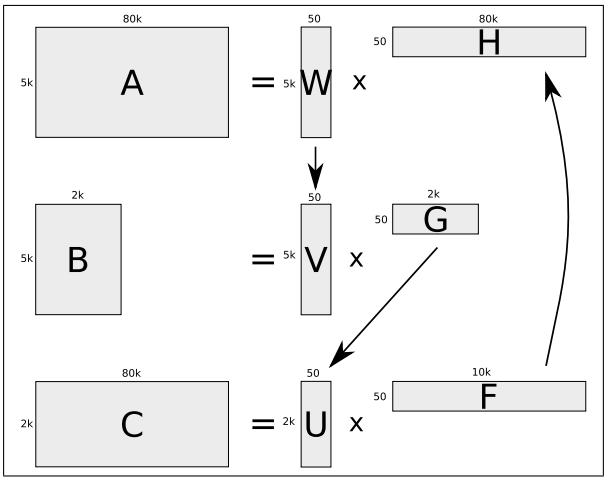

Fig. 1. A graphical representation of the extended NMF

Next, the factorization that is thus created is used for word sense induction. The intuition is that a particular dimension of an ambiguous word is 'switched off', to reveal possible other senses of the word. Matrix H indicates the importance of each syntactic relation given a semantic dimension. With this knowledge, the syntactic relations that are responsible for a certain dimension can be subtracted from the original noun vector. This is done by scaling down each feature of the original vector according to the load of the feature on the subtracted dimension.

The last step is to determine which dimension(s) are responsible for a certain sense of the word. In order to do so, the method is embedded in a clustering approach. First, a specific word is assigned to its predominant sense (i.e. the most similar cluster). Next, the dominant semantic dimension(s) for this cluster are subtracted from the word vector, and the resulting vector is fed to the clustering algorithm again, to see if other word senses emerge. The dominant semantic dimension(s) can be identified by 'folding in' the cluster centroid into the factorization.

A simple k-means algorithm is used to compute the initial clustering. k-means yields a hard clustering, in which each noun is assigned to exactly one (dominant) cluster. In the second step, it is determined for each noun whether it can be assigned to other, less dominant clusters. First, the salient dimension(s) of the centroid to which the noun is assigned are determined. The centroid of the cluster is computed by averaging the frequencies of all cluster elements except for the target word we want to reassign, and weighting the resulting vector with pointwise mutual information [20]. After subtracting the salient dimensions from the noun vector, it is checked whether the vector is reassigned to another cluster centroid. If this is the case, (another instance of) the noun is assigned to the cluster, and the second step is repeated. If there is no reassignment, we continue

with the next word. The target element is removed from the centroid to make sure that only the dimensions associated with the sense of the cluster are subtracted. When the algorithm is finished, each noun is assigned to a number of clusters, representing its different senses.

We use two different methods for selecting the final number of candidate senses. The first method, NMF_{con}, takes a conservative approach, and only selects candidate senses if – after the subtraction of salient dimensions – another sense is found that is more similar to the adapted noun vector. The second method, NMF_{lib}, is more liberal, and also selects the next best cluster centroid as candidate sense until a certain similarity threshold ϕ is reached. Experimentally (examining the cluster output), we set $\phi = 0.2$.

Sense disambiguation The sense inventory that results from the induction step can now be used for the disambiguation of individual instances as follows. For each instance of the target noun, we extract its context words, i.e. the words that co-occur in the same paragraph, and represent them as a frequency vector. Using matrix G from our factorization model (which represents context words by semantic dimensions), this co-occurrence vector can be 'fold in' into the semantic space, thus representing the probability of each semantic dimension for the particular instance of the target noun. Likewise, the candidate senses of the noun (represented as centroids) can be folded into our semantic space using matrix H, which represents the dependency relations by semantic dimensions. This yields a probability distribution over the semantic dimensions for each centroid. As a last step, we compute the Kullback-Leibler divergence between the context vector and the candidate centroids, and select the candidate centroid that yields the lowest divergence as the correct sense.

Example Let us clarify the process with an example for the noun *chip*. The sense induction algorithm finds the following candidate senses:

1. *cache*, *CPU*, *memory*, *microprocessor*, *processor*, *RAM*, *register*
2. *bread*, *cake*, *chocolate*, *cookie*, *recipe*, *sandwich*
3. *accessory*, *equipment*, *goods*, *item*, *machinery*, *material*, *product*, *supplies*

Each candidate sense is associated with a centroid (the average frequency vector of its members), that is fold into the semantic space, which yields a 'semantic fingerprint', i.e. a distribution over the semantic dimensions. For the first sense, the 'computer' dimension will be the most important. Likewise, for the second and the third sense the 'food' dimension and the 'manufacturing' dimension will be the most important.[1]

Let us now take a particular instance of the noun *chip*, such as the one in (1).

[1] In the majority of cases, the induced dimensions indeed contain such clear-cut semantics, so that the dimensions can be rightfully labeled as above.

(1) An N.V. Philips **unit** has **created** a **computer system** that **processes video images** 3,000 times faster than conventional **systems**. Using **reduced instruction - set computing**, or RISC, chips made by Intergraph of Huntsville, Ala., the **system** splits the **image** it '**sees**' into 20 **digital representations**, each **processed** by one *chip*.

Looking at the context of the particular instance of *chip*, a context vector is created which represents the semantic content words that appear in the same paragraph (the extracted content words are printed in boldface). This context vector is again folded into the semantic space, yielding a distribution over the semantic dimensions. By selection the lowest Kullback-Leibler divergence between the semantic probability distribution of the target instance and the semantic probability distributions of the candidate senses, the algorithm is able to induce the 'computer' sense of the target noun *chip*.

Implementational details The SemEval training set has been part of speech tagged and lemmatized with Stanford Part-Of-Speech Tagger [21, 22], and parsed with MaltParser [23] trained on sections 2-21 of the Wall Street Journal section of the Penn Treebank extended with about 4000 questions from the QuestionBank[2] in order to extract dependency triples. The SemEval test set has only been tagged and lemmatized, as our disambiguation model did not use dependency triples as features (contrary to our induction model).

The three matrices needed for our factorization model were constructed using the 5K nouns, 80K dependency relations, and 2K context words (excluding stop words) with highest frequency in the training set, which yields matrices of 5K nouns $\times$ 80K dependency relations, 5K nouns $\times$ 2K context words, and 80K dependency relations $\times$ 2K context words. For our initial k-means clustering, we cluster the 5K nouns into 600 clusters.

The sense induction and disambiguation algorithms were implemented in Python. The interleaved NMF factorization model itself was implemented in Matlab, using 50 iterations, and factorizing the model to 50 dimensions.

3 Word sense induction evaluation in SemEval 2010

3.1 Training and evaluation datasets

Our WSI algorithm is trained and tested on the dataset of the SemEval-2010 WSI&D task [4]. The main difference of this task from the SemEval-2007 WSI task [5] is that the training and testing data are treated separately, which allows for a more realistic evaluation of the clustering models. Word senses are induced from the training data while testing data are used for tagging new instances of the words with the previously discovered senses.

The SemEval-2010 WSI&D task is based on a dataset of 100 target words, 50 nouns and 50 verbs. For each target word, a *training* set is provided from which

[2] `http://maltparser.org/mco/english_parser/engmalt.html`

the senses of the word have to be induced without using any other resources. The training set for a target word consists of a set of target word instances in context (sentences or paragraphs). In this paper, we will focus on the set of nouns, that consists of 716,945 instances.

The senses induced during training are used for disambiguation in the *testing* phase. In this phase, the systems are provided with a testing dataset that consists of unseen instances of the target words. The testset comprises 5,285 noun instances. The instances in the testset are tagged with OntoNotes senses [24]. The systems need to disambiguate these instances using the senses acquired during training.

3.2 Supervised and unsupervised evaluation

The results of the systems participating in the SemEval-2010 WSI&D task are evaluated both in a supervised and in an unsupervised manner. In the *supervised* evaluation, one part of the testing dataset is used as a *mapping* corpus, which serves to map the automatically induced clusters to gold standard (GS) senses, and the other part as an *evaluation* corpus, used to evaluate the methods in a standard WSD task. The mapping between clusters and GS senses serves to tag the evaluation corpus with GS tags.

In the *unsupervised* evaluation, the induced senses are evaluated as clusters of examples (*tw* contexts) which are compared to the sets of examples tagged with the GS senses (corresponding to classes). So, if the testing dataset of a *tw* comprises a number of instances, these are divided into two partitions : a set of automatically generated clusters and a set of GS classes. A number of these instances will be members of both one GS class and one cluster. Consequently, the quality of the proposed clustering solution is evaluated by comparing the two groupings and measuring their similarity.

3.3 Evaluation measures

The supervised evaluation in the SemEval-2010 WSI&D task follows the scheme employed in the SemEval-2007 WSI task [5], with some modifications. The induced senses (clusters) are mapped to GS senses using a mapping corpus, which is a part of the testing sense-tagged dataset. Then, the evaluation corpus, which corresponds to the rest of the testing dataset, is used to evaluate WSI methods in a standard WSD task. The evaluation is performed according to the precision and recall measures employed for the evaluation of supervised WSD systems.

Two evaluation metrics are employed during the unsupervised evaluation in order to estimate the quality of the clustering solutions, the *V-measure* [25] and the *paired F-Score* [26]. *V-Measure* assesses the quality of a clustering by measuring its *homogeneity* (h) and its *completeness* (c). Homogeneity refers to the degree that each cluster consists of data points primarily belonging to a single GS class, while completeness refers to the degree that each GS class consists of data points primarily assigned to a single cluster. V-Measure is the harmonic mean of h and c.

$$VM = \frac{2 \cdot h \cdot c}{h + c} \tag{1}$$

In the *paired F-Score* [26] evaluation, the clustering problem is transformed into a classification problem [4]. A set of instance pairs is generated from the automatically induced clusters, which comprises pairs of the instances found in each cluster. Similarly, a set of instance pairs is created from the GS classes, containing pairs of the instances found in each class. *Precision* is then defined as the number of common instance pairs between the two sets to the total number of pairs in the clustering solution (cf. formula 2). *Recall* is defined as the number of common instance pairs between the two sets to the total number of pairs in the GS (cf. formula 3). Precision and recall are finally combined to produce the harmonic mean (cf. formula 4).

$$P = \frac{|F(K) \cap F(S)|}{|F(K)|} \tag{2}$$

$$R = \frac{|F(K) \cap F(S)|}{|F(S)|} \tag{3}$$

$$FS = \frac{2 \cdot P \cdot R}{P + R} \tag{4}$$

The obtained results are also compared to two baselines. The Most Frequent Sense (*MFS*) baseline groups all testing instances of a *tw* into one cluster. The *Random* baseline randomly assigns an instance to one of the clusters.[3]. This baseline is executed five times and the results are averaged.

4 Evaluation results

4.1 Unsupervised evaluation

In table 1, we present the performance of a number of algorithms on the V-measure. We compare our V-measure scores with the scores of the best-ranked systems in the SemEval 2010 WSI&D task. The second column shows the number of clusters induced in the test set by each algorithm. The *MFS* baseline has a V-Measure equal to 0, since by definition its completeness is 1 and homogeneity is 0.

NMF_{con} – our model that takes a conservative approach in the induction of candidate senses – does not beat the random baseline. NMF_{lib} – our model that is more liberal in inducing senses – reaches better results. With 13.5%, it scores similar to other algorithms that induce a similar average number of clusters, such as Duluth-WSI [27].

Pedersen [27] has shown that the V-Measure tends to favour systems producing a higher number of clusters than the number of GS senses. This is reflected in the scores of our models as well.

[3] The number of clusters of *Random* was chosen to be roughly equal to the average number of senses in the GS.

	VM (%)	#Cl
UoY	20.6	11.54
Hermit	16.7	10.78
KSU KDD	18.0	17.5
NMF_{lib}	13.5	5.42
Duluth-WSI	11.4	4.15
Random	4.2	4.00
NMF_{con}	3.9	1.58
MFS	0.0	1.00

Table 1. V-measure for SemEval noun testset

In table 2, the paired F-Score of a number of algorithms is given. The paired F-Score penalizes systems when they produce a higher number of clusters (low recall) or a lower number of clusters (low precision) than the GS number of senses. We again compare our results with the scores of the best-ranked systems in the SemEval 2010 WSI&D task.

	FS (%)	#Cl
MFS	57.0	1.00
Duluth-WSI-SVD-Gap	57.0	1.02
NMF_{con}	54.6	1.58
NMF_{lib}	42.2	5.42
Duluth-WSI	37.1	4.15
Random	30.4	4.00

Table 2. Paired F-score for SemEval noun testset

NMF_{con} reaches a score of 54.6%, which is again similar to other algorithms that induce the same average number clusters. NMF_{lib} scores 42.2%, indicating that the algorithm is able to retain a reasonable F-Score while at the same time inducing a significant number of clusters. This especially becomes clear when comparing its score to the other algorithms.

4.2 Supervised evaluation

Table 3 shows the recall of our algorithms in the supervised evaluation, again compared to other algorithms evaluated in the SemEval 2010 WSI&D task.

NMF_{lib} gets 57.3% and NMF_{con} reaches 54.5%, which again indicates that our algorithm is in the same ballpark as other algorithms that induce a similar average number of senses.

	SR (%)	#S
UoY	59.4	1.51
NMF_{lib}	57.3	1.93
Duluth-WSI	54.7	1.66
NMF_{con}	54.5	1.21
MFS	53.2	1.00
Random	51.5	1.53

Table 3. Supervised recall for SemEval noun testset, 80% mapping, 20% evaluation

5 Conclusion and future work

In this paper, we presented a quantitative evaluation of a global approach to word sense induction, and compared it to more prevailing local approaches to word sense induction, that induce senses on a per-word basis. The results indicate that the global approach performs equally well, reaching similar results to the state-of-the-art performance of local approaches. Moreover, the global approach is able to reach similar performance on an evaluation set that is tuned to fit the needs of local approaches. The evaluation set contains an enormous amount of contexts for only a small number of target words, favouring methods that induce senses on a per-word basis. The global approach is likely to induce a more balanced sense inventory using a more balanced, unbiased corpus, and is likely to outperform local methods when such an unbiased corpus is used as input. We therefore think that a global approach to word sense induction, such as the one presented here, provides a genuine and powerful solution to the problem at hand, and deserves further attention.

We conclude with some issues for future work. First of all, we would like to evaluate the approach presented here using a more balanced an unbiased corpus, and compare its performance on such a corpus to local approaches. Secondly, we would also like to include grammatical dependency information in the disambiguation step of the algorithm. For now, the disambiguation step only uses a word's context words; enriching the feature set with dependency information is likely to improve the performance of the disambiguation.

Acknowledgments

This work is supported by the Scribo project, funded by the French "pôle de compétitivité" System@tic, and by the French national grant EDyLex (ANR-09-CORD-008).

References

1. Ide, N., Wilks, Y.: Making sense about sense. In: In E. Agirre and P. Edmonds (eds.), Word Sense Disambiguation, Algorithms and Applications, Springer (2007) 47–73
2. Harris, Z.: Distributional structure. Word (1954) 146–162
3. Manandhar, S., Klapaftis, I.P.: Semeval-2010 task 14: Evaluation setting for word sense induction & disambiguation systems. In: Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions, Boulder, Colorado (2009) 117–122
4. Manandhar, S., Klapaftis, I.P., Dligach, D., Pradhan, S.: Semeval-2010 task 14: Word sense induction &disambiguation. In: Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, Uppsala, Sweden (2010) 63–68
5. Agirre, E., Soroa, A.: Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In: Proceedings of the 4th International Workshop on Semantic Evaluations, ACL, Prague, Czech Republic (2007) 7–12
6. Miller, G., Charles, W.: Contextual correlates of semantic similarity. Language and Cognitive Processes **6** (1991) 1–28
7. Navigli, R.: Word sense disambiguation: a survey. ACM Computing Surveys **41** (2009) 1–69
8. Schütze, H.: Automatic word sense discrimination. Computational Linguistics **24** (1998) 97–123
9. Purandare, A., Pedersen, T.: Word sense discrimination by clustering contexts in vector and similarity spaces. In: Proceedings of the Conference on Computational Natural Language Learning (CONLL'04), Boston, MA (2004) 41–48
10. Pedersen, T., Bruce, R.: Distinguishing word senses in untagged text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Providence, RI (1997) 197–207
11. Bordag, S.: Word sense induction: Triplet-based clustering and automatic evaluation. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Trento, Italy (2006) 137–144
12. Schutze, H.: Automatic word sense discrimination. Computational LInguistics **24** (1998) 97–123
13. Widdows, D., Dorow, B.: A graph model for unsupervised lexical acquisition. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING), Taipei, Taiwan (2002) 1093–1099
14. Véronis, J.: Hyperlex: lexical cartography for information retrieval. Computer Speech & Language **18** (2004) 223–252
15. Agirre, E., Martínez, D., de Lacalle, O.L., Soroa, A.: Two graph-based algorithms for state-of-the-art wsd. In: Proceedings of the Empirical Methods in Natural Language Processing (EMNLP) Conference, Sydney, Australia (2006) 585–593
16. Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL98), Volume 2, Montreal, Quebec, Canada (1998) 768–774
17. Pantel, P., Lin, D.: Discovering word senses from text. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada (2002) 613–619
18. Van de Cruys, T.: Using three way data for word sense discrimination. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester (2008) 929–936

19. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems (2000) 556–562
20. Church, K.W., Hanks, P.: Word association norms, mutual information & lexicography. Computational Linguistics **16** (1990) 22–29
21. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000). (2000) 63–70
22. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of HLT-NAACL 2003. (2003) 252–259
23. Nivre, J., Hall, J., Nilsson, J.: Maltparser: A data-driven parser-generator for dependency parsing. In: In Proc. of LREC-2006. (2006) 2216–2219
24. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: Ontonotes: the 90% solution. In: Proceedings of NAACL, Companion Volume: Short Papers on XX. (2006) 57–60
25. Rosenberg, A., Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: Proceedings of the Joint 2007 Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic (2007) 410–420
26. Artiles, J., Amigó, E., Gonzalo, J.: The role of named entities in web people search. In: In Proceedings of EMNLP. (2009) 534–542
27. Pedersen, T.: Duluth-wsi: Senseclusters applied to the sense induction task of semeval-2. In: Proceedings of the 5th International Workshop on Semantic Evaluation, Uppsala, Sweden, Association for Computational Linguistics (2010) 363–366