Using a Maximum Entropy Classifier to link "good" corpus examples to dictionary senses

Alexander Geyken[1], Christian Pölitz[2], Thomas Bartz[2]

[1] Berlin-Brandenburg Academy of Sciences, Jägerstr. 22/23, D-10117 Berlin, Germany
[2] Technische Universität Dortmund, Fakultät für Informatik, Otto-Hahn-Str. 14, 44227 Dortmund, Germany
E-mail: geyken@bbaw.de, {poelitz,bartz}@tu-dortmund.de

Abstract

A particular problem of maintaining dictionaries consists of replacing outdated example sentences by corpus examples that are up-to-date. Extraction methods such as the good example finder (GDEX; Kilgarriff, 2008) have been developed to tackle this problem. We extend GDEX to polysemous entries by applying machine learning techniques in order to map the example sentences to the appropriate dictionary senses. The idea is to enrich our knowledge base by computing the set of all collocations and to use a maximum entropy classifier (MEC; Nigam, 1999) to learn the correct mapping between corpus sentence and its correct dictionary sense. Our method is based on hand labeled sense annotations. Results reveal an accuracy of 49.16% (MEC) which is significantly better than the Lesk algorithm (31.17%).

Keywords: WSD; maximum entropy; collocations; legacy dictionaries; example sentences

1. Introduction

Keeping dictionaries up-to-date is a very time consuming task that involves regular checks throughout the entire dictionary for all types of lexicographic information. One particular problem consists of replacing outdated example sentences in the dictionary by suitable corpus examples that are up-to-date or of adding corpus examples to new entries. In general today's corpora of several billion words of text are too large to allow for regular manual inspection of the entire set of frequent words. Indeed, Moon (2007) states that the 25,000 most frequent words in English all have frequencies higher than one per million tokens. For a one billion word corpus this would amount to analysing 1,000 corpus hits. Since many of today's corpora exceed 10 billion words, this would quickly result in numbers that are no longer feasible within the budget and time constraints of today's lexicographic projects. Several methods to automate this task have been developed, the most popular being the "good" example finder (GDEX; Kilgarriff et al., 2008). GDEX is a rule based software tool that suggests "good" corpus examples to the lexicographer according to predefined criteria such as sentence length or word frequency, or lexicogrammatical criteria such as the presence/absence of pronouns or named entities. The goal of GDEX is to reduce the number of corpus examples to be inspected by extracting only the n-"best" examples. The ideas of

GDEX have been used for languages other than English (Kosem et al., 2011, for Slovene) and have given rise to different implementations (Didakowski et al., 2012, for German; Volodina et al., 2012, for Swedish).

The goal of our work is to extend GDEX to polysemous entries. More precisely we attempt to link a given corpus sentence extracted by GDEX to its appropriate dictionary sense (in the case of a polysemous entry). The method we employ is a machine learning technique (cf. section 3). The main hypothesis of our work is that the results of our machine learning approach improve if the linking is not only performed to a dictionary sense represented by a sense number and a definition but rather on the full dictionary sense. In the case of a large reference dictionary this includes the example sentences, citations and the set phrases.

The remainder of this article is structured as follows. In section 2 we present related work in the field of Word Sense Disambiguation. In section 3 we describe the resources we use. The machine learning approach is described in section 4. We then report on an experiment with 100 polysemous and frequently used German words (section 5). The last section discusses the results and presents some ideas for further research.

2. Word Sense Disambiguation

Word Sense Disambiguation (WSD) plays an important role in Natural Language Processing. Many approaches have been carried out in this area. Starting from the pioneer work of Lesk (1986), automatic methods to assign text examples to possible senses given from a dictionary for instance have become increasingly important. The first approaches for assigning senses to given text examples used pure word overlaps between the text and definitions for the senses. These definitions can be for instance from a dictionary or, as proposed by Vasilescu et al. (2004), from synsets from WordNet. Besides pure word overlaps to assign senses to texts, knowledge based methods have also proven successful. Navigli and Velardi (2005) introduce structural and rule based representations of possible senses to efficiently map them to text examples. More recently, machine learning approaches based on supervised methods have emerged in WSD, including Neural Networks (Moony, 1996), Näve Bayes (Patterson, 2007), Ensemble Methods (Escudero, 2000) and Support Vector Machines (Keok & Ng, 2002). A detailed introduction to WSD and a survey on the different methods to solve it can be found in Navigli (2009).

3. Resources

The resources used for the work presented here are threefold: a dictionary, a large database of collocations and GDEX. All these resources are part of the DWDS (Digitales Wörterbuch der deutschen Sprache, Digital Dictionary of the German Language), a project of the Berlin-Brandenburg Academy of Sciences and Humanities (BBAW). DWDS is a long term project of BBAW. Its goal is to compile a large aggregated word information system based on large legacy dictionaries, large corpora, word statistics and automated methods to speed up the compilation process (Geyken, 2014).

The dictionary used for our work is the large “Wörterbuch der deutschen Gegenwartssprache” (dictionary of the German contemporary language, WDG, www.dwds.de), a synchronic dictionary of 4,800 pages with 120,000 keywords, compiled between 1961 and 1977. The electronic version of the WDG is encoded in TEI. Each entry consists of a form and a sense part; the sense comprises definitions, diasystematic markers, made-up examples and corpus examples. Relevant to our work are the following components of the sense element: definition, examples made-up by the lexicographer and citations from corpora. We will call these components *dictionary sense* in the remainder of this article. An example for the entry *Leiter* (en. leader, ladder, conductor) drawn from the WDG is given in Table 1. Only sense 2 is fully expanded; for senses 1 and 3, definitions only are provided. The full entry can be looked up at the project’s website (www.dwds.de).

Sense 1: *Gerät aus Holz oder Leichtmetall* (en.: device made of wood or light metal)
Sense 2: *jmd., der etw. leitet, an der Spitze von etw. steht* (s.o. who directs sth., who is at the top of sth.) made-up examples and constructions: *ein technischer, kaufmännischer, künstlerischer, staatlicher, kommissarischer Leiter* (a technical, commercial, artistic, governmental, acting director) *der Leiter einer Baustelle, Abteilung, Schule, Delegation, Touristengruppe, Behörde, Expedition, eines Krankenhauses, Unternehmens* (the head of a construction site, department, school, delegation, tourist group, authority, expedition, hospital, company) corpus example: *Heut bin ich im Funk Leiter vom Dienst* (Today I am in the radio manager on duty) [Klepper, J., Schatten, 1960, p. 56]
Sense 3: Stoff, der Energie leitet (substance that passes energy)

Table 1: entry Leiter in the WDG

The second resource is the DWDS-Wortprofil (Didakowski & Geyken 2012), an implementation of the sketch engine (Kilgarriff et al., 2004) for German. DWDS-Wortprofil provides co-occurrence lists for twelve different grammatical relations (Tables 2 and 3) and links them to their corpus contexts. The co-occurrence lists and their ordering are based on statistical computations over a German corpus of currently 1.783 billion tokens. For syntactic annotation the rule based dependency parser SynCoP (Syntactic Constraint Parser; Didakowski, 2008) is used. A grammar for the SynCoP parser was developed which is designed for the specific relation extraction task. Therefore, issues like the attachment of sub-clauses or specific rare syntactic phenomena are not dealt with in this grammar.

syntactic relation	**part-of-speech tuples**
accusative object	{<verb,noun>}
active subject	{<verb,noun>}
adjective attribute	{<noun,adjective>}
coordination	{<verb,verb>,<noun,noun>,<adjective,adjective>}
dative object	{<verb,noun>}
genitive attribute	{<noun,noun>}
modifying adverbial	{<verb,adverb>,<adjective,adverb>}
passive subject	{<verb,noun>}
predicative complement	{<noun,noun>,<noun,adjective>}
verb prefix	{<verb,prefix>}

Table 2: binary relations

syntactic relation	**part-of-speech tuples**
comparative conjunction	{<noun,conjunction,noun>,<verb,conjunction,noun>}
prepositional group	{<noun,preposition,noun>,<verb,preposition,noun>}

Table 3: ternary relations

As a result of the statistical computations, the database contains 11,980,910 distinct co-occurrence pairs (types) with a total of 257,402,167 tokens. The DWDS-Wortprofil is part of the web platform of DWDS and is continually extended with new corpora. In its current version it is possible to query 104,704 different lemma/part-of-speech pairs.

The third resource used for this work is a set of corpus sentences. We use an implementation of GDEX for German (Didakowski et al., 2012) to extract the n-best corpus sentences for a given word. The underlying text corpora for this extraction task are the corpora of the DWDS project. The corpora comprise a total of 4 billion words and consist of four subcorpora: 1) the DWDS-Kernkorpus of the 20^{th}/21^{st} century, a balanced reference corpus of 110 million tokens (Geyken, 2007); 2) a balanced historical corpus currently comprising of 120 million tokens for the period from 1600 to 1900, compiled at the BBAW for the project *Deutsches Textarchiv* (DTA, German Text Archive, www.deutschestextarchiv.de); 3) a corpus of ten influential national daily and weekly newspapers, which currently consists of 3.5 billion tokens in 8 million documents; and 4) several special corpora with a total of 200 million tokens, including a large blog corpus, a corpus of contemporary interviews and a corpus of subtitles.

4. Method

The standard approach by Lesk (1996) to match a text to senses with given definitions is to count the words that both definitions and texts have in common. The higher the number of common words, the more likely that the text will have the corresponding sense. Formally, for a text $t = w_1 \dots w_k \dots w_n$ being the context of a key word w_k, a set of applicable senses $\{s_i\}$ with corresponding definitions $\{d_i = w_1^i \dots w_{m_i}^i\}$, the standard Lesk algorithm calculates the numbers n_i that are the sum of common words from t and d_i. We assign the sense s_j to text t with $n_j = \max_{s_i} n_i$, for all applicable senses s_i. A major drawback of this approach is that for shorter texts and definitions the chance to have overlap decreases.

A simple extension of the Lesk method to lexical databases was proposed by Vasilescu et al. (2004). The authors extend the concept of overlap of words from sense definitions and key word context (i.e. a corpus sentence) to WordNet. A drawback of their approach is that they can only match to WordNet senses and not to arbitrary dictionary entries.

We propose to extend the Lesk algorithm in such a way that we do not only count the number of intersecting words, but also all words that are statistically salient co-occurrences (i.e. with a logDice > 0) in the DWDS Wortprofil, as explained in section 3. These sets of co-occurrences, henceforth called word-profiles, are computed for all content words (nouns, verbs, adjectives and adverbs) of all dictionary senses of a given headword; i.e. the definition, the example sentences and the corpus citations that are part of the legacy dictionary. This results in a mapping from each headword to a list containing all statistically salient co-occurring words from the word profiles together with the corresponding logDice values. The match from a corpus sentence extracted by GDEX to a dictionary sense is performed by matching all word profiles from the content words in the corpus sentence with the dictionary senses. This means, for each word w_i in the corpus sentence and each word w_l^i in a dictionary sense s_j, we count the number of common words in the two corresponding word profiles weighted by the logDice from the word profile of the word from the key word context. Finally, we sum up all aggregated logDices. The "best" dictionary sense for a given corpus sentence is the one that corresponds to the largest sum (compared to the other dictionary senses). This extension of the Lesk algorithm is henceforth called Lesk_{ext}. An example of how Lesk_{ext} is performed on the dictionary example *Leiter* (cf. Table 1 above) is given in Tables 4 and 5. Table 4 illustrates the logDices for the collocations that the two nouns *Spitze* (top) in the dictionary definition and *Verantwortung* (responsibility) in the corpus example have in common. Table 5 displays the total number of collocations as well as the sum of the logDice values for both, sense 1 and sense 2.

	dictionary definition	**Corpus example**
	Leiter, sense 2 "*jmd. der etw. leitet, an der Spitze von etwas steht*" (so. who leads, is in the top position of sth.)	"Aufgabe der HI ist es nicht, den Leitern diese Verantwortung abzunehmen." (It is not the task of the HI, to remove the responsibility from the leaders.)
content words	Spitze	Verantwortung

collocations in common/relation	**logDice**	**e.g. "Spitze"**	**e.g. "Verantwortung"**	**logDice**
adjective attribute	4.72	*international* (international)		5.08
	1.79	*gesellschaftlich* (social)		8.23
	2.93	*alleinig* (sole)		8.87
	Σ 9.44			Σ 22.18
genitive attribute	6.60	*Unternehmen* (enterprise)		5.48
	5.99	*Aufsichtsrat* (directorate)		5.80
	5.29	*Politik* (politics)		6.00
	Σ 17.88			Σ 17.28
predicative complement	1.68	*hoch* (high)		3.60
	3.14	*deutlich* (clear)		3.97
	Σ 4.82			Σ 7.57
		...		

Table 4: Example: Mapping of dictionary examples and corpus sentences
(identical senses: head/leader)

	dictionary example	**corpus sentence**	**logDice (sum)**
sense 2	head/leader	head/leader	798.22
content words (86 collocations in common)	„Spitze" (top position)	„Verantwortung" (responsibility)	
sense 1	ladder	head/leader	62.95
content words (8 collocations in common)	„hoch" (high)	„Verantwortung" (responsibility)	

Table 5: Example: Aggregated logDice values

The DWDS-Wortprofil also specifies the syntactic relation between a word and its co-occurrences. We propose to aggregate the logDice values for co-occurrences from the word profiles as before, but now for each of the syntactic relations individually in order to measure the impact on individual syntactic relation. Thus, we can measure the impact on the type of syntactic relation of the matching process to its corresponding dictionary sense. As mentioned above there are 10 binary relations and two ternary relations in the DWDS-Wortprofil. This means we are not getting a single sum after the match of all word profiles but a vector with the sum of the aggregated logDices for each relation. Next, to assign the best weight to each syntactic relation we use a Maximum Entropy Classifier (Nigam et al., 1999) that models the probability distribution of a given context and a given definition from the senses. Formally, the probability of a sense s for a given corpus sentence t is defined as $p(s|t) = e^{\omega'\varphi(s,t)} / Z$ for a feature vector $\varphi(s,t)$, a weight vector $\omega$and the normalization constant Z. Each feature in $\varphi(s,t)$ is the sum of the logDices of the matching words for the dictionary sense s and (sentence) context t for a relation as explained above. We find the optimal weights ω by maximizing the joint probability over a training set $\{(S_k, T_k)\}$ of key word contexts T_k for a given number of key words $w_k \in K$ with hand labeled senses S_k with given definitions. The optimal ω is the parameter vector that maximizes the log likelihood of our given training data. The resulting optimization problem is defined in the following way:

$$= \operatorname{argmax}\{ \sum_{w_k \in K} \log(S_k | T_k, \omega) = \sum_{(s,t)\in(S_k,T_k)} \log\left(\frac{e^{\omega'\varphi(s,t)}}{\sum_{s'} e^{\omega'\varphi(s',t)}}\right) \}$$

We solve the above optimization problem with a standard BFGS solver (Broyden–Fletcher–Goldfarb–Shanno algorithm) that performs a quasi-Newton optimization as for instance proposed by (Byrd et al., 1995). For the sense association example in Table 4, the MEC classifier provides a probability distribution stating that sense 2 is selected with a probability of 0.9 whereas sense 1 has only a 0.1 chance.

5. Experiment

In an experiment we selected 100 highly polysemous headwords (75 nouns, 25 verbs). These words have a total of 857 fine-grained senses (314 main or coarse-grained senses) in our dictionary (WDG). The list of headwords with English translations of the most prominent sense of the item in parenthesis is the following:

ablösen (supersede), Achse (axis), Adresse (address), Agent (agent), anschließen (connect), Ansicht (view), anstellen (do), Atmosphäre (atmosphere), aufheben (cancel), Aussprache (pronunciation), ausziehen (move out), Bank (bank), beschreiben (describe), Betrieb (operation), Blase (bubble), eingehen (enter), Einheit (unit), Einsatz (use), Eis (ice), eröffnen (open), Fall (case), feststellen (find), Film (movie), finden (find), Flucht (flight), Gehäuse (housing), Gemeinde (community), Gericht (court), Geschichte (history), Grund (reason), handeln (act), Höhe (height), Interesse (interest), Kapelle (chapel), Kasse (checkout), klappen (fold), Kopf (head), Körper (body), kosten (cost), Leder (leather), Lehre (teaching), Leiter (ladder), lesen (read), Mal (time), Mark (marrow), Markt (market), Masche (stitch), Maschine (machine), Messe (fair), Mine (mine), Mission (mission), Moment (moment), Morgen (morning), Mutter (mother), nachsehen (check), Operation (operation), Parkett (parquet), passen (match), passieren (happen), Passion (passion), Pause (pause), Pension (guesthouse), Phase (phase), Piste (runway), Praxis (practice), Probe (sample), Prozess (process), riechen (smell), Rolle (role), Satz (sentence), Schatz (treasure), Scheibe (disc), scheinen (appear), Schloss (castle), Sitz (seat), sitzen (sit), Sohle (sole), Stärke (strength), Stelle (location), Steuer (tax), Stimme (voice), stimmen (vote), streichen (paint), Strom (current), Tafel (blackboard), Theater (theater), Ton (clay), Tonne (ton), Truppe (troops), Verfahren (method), Verfassung (constitution), Verhältnis (relationship), Vermittlung (mediation), versichern (reassure), versprechen (promise), Vorstellung (representation), Welle (wave), Wende (turn), Zelle (cell), zugeben (admit)

For each headword, we extracted 20 sentences using the GDEX method (Didakowski et al., 2012) applied to the DWDS corpora (www.dwds.de). All 2,000 example sentences were manually annotated with their corresponding dictionary senses by two annotators. We randomly split the example sentences into a training set of 750 sentences and test set of 1,250 sentences and we applied the Lesk algorithm and the Maximum Entropy Classifier method, as described in section 4.

6. Results and Discussion

The results of our experiment show that the Maximum Entropy Classifier significantly improves on the Lesk_{ext} algorithm. Both methods were applied on the same training data using the same resources, including the data from the DWDS-Wortprofil. As stated above, we have an average of 8.57 fine-grained senses. Thus, a random selection as base-line would predict an accuracy rate of 11.67%. With the Lesk algorithm based on intersection of co-occurring words of the DWDS-Wortprofil we achieve an accuracy of 31.17% for the test set. The Maximum Entropy Classifier further optimizes Lesk_{ext} by taking into account the specific syntactic relations as well as the weights provided by the logDice values that are used to compute the co-occurrence strength between the headword and its collocate. The application of the Maximum Entropy Classifier provides an accuracy of 49.16% for fine-grained senses in our test set. There are also differences between the accuracy of nouns (51.8%) and verbs (44.24%). The lower accuracy for verbs is due to the fact that the semantic information of the WDG is poorer for verbs, i.e. it frequently uses only placeholders (such as s.o., sth.) in its sense descriptions.

We have also investigated the impact of the sense granularity. As stated above there are 314 coarse-grained senses for our training set. Hence the base-line would predict an accuracy of 31.8%. If applied on coarse-grained senses, the accuracy of the Maximum Entropy Classifier augments by about 7%, i.e. 55.74%, instead of 49.1% for fine-grained senses. Again, there are differences between nouns and verbs: MEC provides an accuracy of 58.69% for nouns but only 46.88% for verbs.

Another result concerns the quality of GDEX that we evaluated indirectly by the inter annotator agreement. For our test set we obtain an inter annotator agreement (IAA) of kappa = 0.78 for fine-grained senses. Kappa for coarse-grained senses rises by 7% to arrive at 0.85. These kappa values seem high compared to other WSD tasks. One reason for this finding may be that the examples extracted by our GDEX extractor are more homogeneous than a selection by "chance". Indeed, for our data we found that the main sense (that occurs most frequently) is attributed to an average of about 11 out of 20, i.e. 55% (± 2% standard deviation), of the examples for each headword. The second most frequent senses cover only about four to five examples (22.4% ± 1.2%); the other senses even fewer (0–2 examples, 9.8% ± 0.56%). The observation that regular senses might be overweighted by GDEX is shared e.g. by Cook et al. (2014: 320) who claim that "example-finding software does not yet routinely achieve the contextual diversity that characterizes example-sets selected by skilled lexicographers."

Although our MEC improves on the Lesk algorithm it still does not improve to the base-line of always taking the main sense, which in the case of our dictionary consists of the 1st sense. The lines of improvement concern two areas: we plan to enrich the knowledge base with paradigmatic information from the German WordNet (GermaNet, Kunze & Lemnitzer, 2002). Furthermore, we can expect the results of our method to improve with the amount of available example sentences in the dictionary senses. Indeed, example sentences are underrepresented in the WDG as this dictionary was compiled before the era of electronic corpora. Therefore, we plan to repeat our experiments on the basis of the Duden dictionary (Duden-GWDS 1999). Duden has significantly more corpus examples. In the coming months, the Maximum Entropy Classifier will be integrated as a web service in the infrastructure of the Dictionary Writing System of the DWDS project.

7. Acknowledgements

This research has been carried out in the context of the BMBF-funded project KobRA (Korpusbasierte Recherche und Analyse mit Hilfe von Data-Mining, grant ID 01UG1245).

8. References

Byrd, R. H., Lu, P., Nocedal, J. & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal of Scientific Computing* 16, 5 (September 1995), pp. 1190-1208.

Cook, P., Rundell, M., Lau, J. H. & Baldwin, T. (2014). Applying a Word-sense

Induction System to the Automatic Extraction of Diverse Dictionary Examples. In: *Proceedings EURALEX*, Bolzano, Italy.

Didakowski, J. (2008a). ‘Local Syntactic Tagging of Large Corpora Using Weighted Finite State Transducers’. In A. Storrer et al. (eds.), *Text Resources and Lexical Knowledge: Selected Papers from the 9th Conference on Natural Language Processing. KONVENS 2008*. Berlin: Mouton de Gruyter, pp. 65-78.

Didakowski, J., Geyken, A. & Lemnitzer, L. (2012). Automatic example sentence extraction for a contemporary German dictionary. In: *Proceedings EURALEX*, Oslo.

Didakowski, J. & Geyken, A. (2012). From DWDS corpora to a German Word Profile – methodological problems and solutions. In: *Network Strategies, Access Structures and Automatic Extraction of Lexicographical Information. 2. Arbeitsbericht des wissenschaftlichen Netzwerks „Internetlexikografie“. In Online publizierte Arbeiten zur Linguistik 2/2012 (OPAL)*, Mannheim: Institut für deutsche Sprache, pp. 43-52.

Duden-GWDS (1999). *Das große Wörterbuch der deutschen Sprache.* 10 volumes. Mannheim/Leipzig/Wien/Zürich: Dudenverlag.

Escudero, G., Màrquez, L. & Rigau, G. (2000), 'Boosting Applied to Word Sense Disambiguation', In *Proceedings of the 12th European Conference on Machine Learning, ECML.* Barcelona, Catalonia. 2000.

Geyken, A. (2014). Methoden bei der Wörterbuchplanung in Zeiten der Internetlexikographie. In U. Heid, S. Schierholz, W. Schweickard, H. E. Wiegand, R. Gouws, & W. Wolski (eds.). *Lexicographica.* Berlin / Boston: de Gruyter, S. pp. 77-112.

Keok, Y.-L. & Ng, H.-T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10* (EMNLP '02), Vol. 10. Association for Computational Linguistics, Stroudsburg.

Kilgarriff, A., Rychly, P., Smrz, P. & Tugwell, D. (2004). The Sketch Engine. In: G. Williams & S. Vessier (eds.). *Proceedings of the XI. Euralex Conference.* Lorient: Université de Bretagne, pp. 105–116.

Kilgarriff, A., Husák, M, McAdam, K., Rundell, M & Rychlý, P. (2008): GDEX - Automatically Finding Good Dictionary Examples in a Corpus. In: *Proceedings of the XIII EURALEX International Conference.* Barcelona: Universitat Pompeu Fabra, pp. 425-433.

Kosem, I., Husák, M., & McCarthy, D. (2011). ‘GDEX for Slovene’. In I. Kosem & K. Kosem (eds.) *Electronic Lexicography in the 21st Century: New applications for new users, Proceedings of eLex 2011.* Ljubljana: Trojina, Institute for Applied Slovene Studies, pp. 151-159.

Kunze, C. & Lemnitzer, L. (2002). GermaNet - representation, visualization, application. In: *Proceedings of LREC 2002, main conference, Vol. V.*, pp. 1485-1491.

Lau, J. H., Cook, P., McCarthy, D., Gella, S., & Baldwin, T. (2014). Learning Word

Sense Distributions, Detecting Unattested Senses and Identifying Novel Senses Using Topic Models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, Baltimore, USA.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation.* New York, NY, USA. ACM, pp. 24-26.

Mooney, R. J. (1996). 'Comparative Experiments on Disambiguating Word Senses: An Illustration of the Role of Bias in Machine Learning'. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-96)*, Philadelphia, PA, pp. 82-91.

Navigli, R. (2009), Word Sense Disambiguation: a Survey. *ACM Computing Surveys, 41(2)*, ACM Press, pp. 1-69.

Navigli, R. & Velardi, P. (2005), 'Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation', *IEEE Transaction on Pattern Analysis and Machine Intelligence* 27, pp. 671-674.

Nigam, K, Lafferty, J & McCallum, J. (1999). Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering.*

Pedersen, T. (2007). 'Learning Probabilistic Models of Word Sense Disambiguation', *CoRR* abs/0707.3972 .

Rychly, P. (2008). A lexicographer-friendly association score. In P. Sojka & A. Horák, (eds.) *Proceedings of Second Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2008.* Brno: Masaryk University, pp. 6-9.

WDG [1961-1977]: Wörterbuch der deutschen Gegenwartssprache in 6 volumes (4,800 pages). Akademie-Verlag: Berlin

Vasilescu, F.; Langlais, P. & Lapalme, G. (2004). Evaluating Variants of the Lesk Approach for Disambiguating Words. In 'LREC' , European Language Resources Association.

Volodina, E., Johansson, R. & Johansson-Kokkinakis, S. (2012). Semi-automatic selection of best corpus examples for Swedish: initial algorithm evaluation. *Workshop on NLP in Computer-Assisted Language Learning. Proceedings of the SLTC 2012 workshop on NLP for CALL.* Linköping Electronic Conference Proceedings 80, pp. 59–70.