Discrimination of Word Senses with Hypernyms

Artem Revenko and Victor Mireles

Semantic Web Company, Vienna, Austria
{artem.revenko,victor.mireles-chavez}@semantic-web.com

Abstract. Languages are inherently ambiguous. Four out of five words in English have more than one meaning. Nowadays there is a growing number of small proprietary thesauri used for knowledge management for different applications. In order to enable the usage of these thesauri for automatic text annotations, we introduce a robust method for discriminating word senses using hypernyms. The method uses collocations to induce word senses and to discriminate the thesaural sense from the other senses by utilizing hypernym entries taken from a thesaurus. The main novelty of this work is the usage of hypernyms already at the stage sense induction. The hypernyms enable us to cast the task to a binary scenario, namely teasing apart thesaural senses from all the rest. The introduced method outperforms the baseline and has indicates accuracy above 80%.

Keywords: thesaurus, controlled vocabulary, word sense induction, entity linking, named entity disambiguation

1 Introduction

Information retrieval can successfully provide services like search or recommendation only once different senses in the corpus are distinguished. Studies show that in the everyday usage of English about 80% of words are ambiguous [19]. Even when controlled vocabularies are available, it is often the case that a label representing a concept has "non-technical" senses, and these senses are also present in the given corpus at hand. Thus, the task of word sense disambiguation (WSD) is still called for in the presence of controlled vocabularies.

Controlled vocabulary refer here to a finite, well specified set of *concepts* that typify a specific domain. Each concept has an associated set of *labels*. A label can be a word in a broad sense, i.e. it may be word or words habitually used in a (natural) language, an acronym or even an arbitrary sequence of symbols taken from a chosen alphabet. A controlled vocabulary can thus be used to capture synonymy, by grouping different synonyms as labels of the same concept. Usually we represent a concept by one of its labels that is chosen in advance and is called the *preferred label*. Since controlled vocabularies express no semantic relationship between concepts their use in disambiguating senses is limited.

In order to enrich controlled vocabularies with semantic information, often the mutually inverse relations of *hyponym* vs. *hypernym* are considered. The hypernym of a concept x is defined as a concept whose meaning includes the meaning of x, so that any concrete entity that is an instance of concept x is also an instance of its hypernym. We call a *thesaurus* a tuple consisting of a vocabulary and list of hypernym/hyponym relations between its concepts.

In particular, thesauri encoding these relations via the *skos:narrower* and *skos:broader* predicates following the standardized SKOS vocabulary [3] are commonly found in the industry [15, p. 169][1], in part because they can be built with little effort for domain-specific use-cases. This is in contrast to more general knowledge graphs which can aid in disambiguating senses (often at high computational cost [2]), but notably they are also costly to compile, and thus not widely adopted.

In order to leverage the knowledge encoded in a thesaurus for tackling WSD, we begin by noting the following:

- Ambiguity is a characteristic of words: concepts are non-ambiguous. Thus, it is enough to map a word onto a concept in order to disambiguate its sense. Many concepts, however, including those not present in a thesaurus, may have the same label.

  Example 1. The STW thesaurus [4] includes a concept with the label *Bond* (`http://zbw.eu/stw/descriptor/12234-1`),
  which is hyponym of a concept with the label *Securities* and is a hypernym of a concept with the label *Asset-backed securities*. Yet, in the phrase *We invite our top investors to* bond *while playing golf*, the word *bond* does not refer to the thesaural concept.

- In the case of domain-specific analysis of texts, it is sufficient to determine whether a given word is being used in the sense encoded in the thesaurus, referred to as the (*thesaural sense*), or not. Casting this task to a binary task also simplifies it. Consequently, we are not interested to find all the possible senses and disambiguate them. Rather, we aim at disambiguating correctly only one chosen (thesaural) sense.

1.1 Problem Statement

We present in this work a method that tackles the domain-specific case of WSD. It is noteworthy that this method does not require all possible senses of a word to be contained in the thesaurus. This makes it specially useful in the industrial environment, where usually only small thesauri are available.

Specifically, we take as an input a corpus, a thesaurus, and a concept from the thesaurus, one of whose labels is found throughout the corpus. We call this concept the *target entity*. The problem that we deal with is to distinguish, for each document in the corpus, whether the target entity is used in the thesaural sense or not. Thus, the end result is a partition of the corpus into two disjoint collections: “this” and “other”. The collection “this” contains the documents that feature the target entity in the thesaural sense.

[1] Though strictly speaking the semantics of broader/narrower relation can be more general than hypernym/hyponym, for example, the meronym relation can also be encoded as broader/narrower.

The contributions of this work are the following:

- We introduce a method for Word Sense Induction (WSI) with the usage of hypernyms.
- We introduce a pipelined work-flow to discriminate between thesaural and non-thesaural senses of the target entity, by utilizing its hypernyms.
- We prepare and carry out an experiment that resembles a real world use case: concepts have multiple labels and the corpus is ambiguous.

2 Related Work

The problem of word sense disambiguation has attracted much attention for several decades now. We refer the reader to [11] for a review. Here we would like to mention that one could roughly distinguish three classes of approaches to solve WSD: supervised, unsupervised and knowledge-based. The unsupervised methods are usually characterized by

- No need for external knowledge, therefore, the system is broadly applicable. However, the results depend a lot on the corpus.
- No expected user involvement.

Our method makes use of the knowledge in thesaurus, in this sense it is knowledge-based. However, it shares many features of the unsupervised methods in that it does not expect any information about the non-thesaural senses (even their number) and that no user involvement is expected.

Among the large amount of research that has been done on this problem, there is some work that is tightly connected to the approach presented in this paper, we describe it in the subsequent subsections.

2.1 Knowledge-Based Methods a.k.a. Named Entity Linking

The knowledge-based methods [18, 17] are based on large static knowledge graphs (KG) that are computed a priori, usually with great effort. Such a graph can be, for example, WordNet [12, 23]. This graphs are assumed to include all senses of the target word, either explicitly as labels of the nodes or not. A corpus is then analyzed and, depending on it, the KG is traversed (e.g. with a random walk [1]) and the possible senses are thus discovered. It is important to point out that using a KG exclusively, without aid of the corpus, is known to lead to poor results, specially in domain-specific cases in which the thesaural sense is unlikely to be part of the knowledge base.

In the problem statement we focus on a single target entity. One can naturally extend the method and run it for all the identified named entities to link it to the correct sense in the KG. The method would benefit from the hypernyms contained in the KG. However, our method makes use of the corpus, therefore it may not be appropriate for disambiguating very short strings, like search queries.

2.2 Word Sense Induction

The task of WSI is a preliminary step for tackling WSD This task consists of finding (inducing) the senses present in unannotated data. In the terms described above, the input to this task is a corpus and a target word. The outcome is an enumeration of all the senses found for the target word.

Of the many ways the WSI problem has been approached, it is important to mention two in the context of this work. The first is the so-called sense embedding [13, 9]. These methods follow the distributional assumption according to which similar words appear in similar contexts [10], and they apply skip-gram models to reduce the dimensionality of related words and assign them to a given word as its 'semantic signature'. These context embeddings are then clustered to determine the different senses of each word. We should mention that, in our experience, these methods fail in cases where two senses of a word lead to very similar contexts, such as "Americano" which can refer to either a cocktail or a coffee.

The second common approach to the WSI task which is of interest here, is to analyze the text and extract from it collocation graphs: graphs whose nodes are words found in the text close to the target word, and whose edges are weighted to reflect the strength of this collocation. This has the advantage that previously unknown senses can be induced and described with the help of collocations. To weight the edges of the graph, conditional probabilities [21], Dice scores [5] or word co-occurrences relative to a reference corpus [8] have been used. More than that, discrete features derived from the syntactic use of each word (e.g. [16]) may be used. In the current paper we abstain from relying on any language specific tools such as part of speech taggers or sentence parsers in order to stay language independent. We make use of stemmers, however, they only help to improve results and are not essential for the introduced methods. Moreover, stemmers are one of the simplest language specific tools and are widely available for many languages.

The next step in the collocation-based graph algorithms is to identify a collection of sets of nodes, each of which correspond to a sense. Once done, WSD can be carried out by clustering the contexts where the target word appears. There have been several approaches to identifying these senses. Graph clustering via various algorithms has been a popular approach (e.g. in [7, 5, 16]). Another approach, built specifically for inducing senses, is HyperLex [21], which defines senses as hubs (highly connected nodes) in the collocation graph along with their immediate neighbors. To identify senses, HyperLex sorts the nodes of the collocation graph by degrees. Senses are induced by taking and removing one by one the hubs from list, along with their immediate neighbors. In this paper we use a variant of HyperLex introduced in [5] with PageRank scores in place of degrees.

Other works, such as [16], use hypernyms after the senses are already induced. This work, in contrast, takes hypernyms into account already at the stage of graph clustering. That way we guarantee that

- the thesaural meaning of the target word is captured in a single sense and

– the level of granularity of the sense is defined by the structure of the thesaurus.

We note that in the case of collocation graphs, there is no explicit description of what a particular sense is. This can be contrasted with the discussion in the previous section, in which senses are defined as concepts in a thesaurus with known hypernym/hyponym relations; or with the static KG based approach to LSI, in which senses are pre-existing entities in the graph. This lack of interpretability of a sense is overcome in this work by relating at least one of the collocation-derived senses with a concept in the thesaurus.

3 Method

In this section we introduce a method to solve the task introduced in Section 1.1. We rely on the "one sense per document" assumption [22] and on the assumption that the sense of the entity can be unambiguously deduced from its hypernyms or more generally, a thesaurus.

Our method consists of 2 steps:

1. WSI; the outcome is a set of senses with one distinguished thesaural sense.
2. WSD, i.e. classification of each occurrence of the target word into one of the senses.

3.1 WSI with Hypernyms

As already mentioned in the introduction, for the WSI task we use HyperLex [21], implementing it in a similar way to [5]. However, we also introduce hypernyms in the WSI process, therefore we will denote the new modification as *HyperHyperLex*. The process is presented in Figure 1.

In the **first** step we compute the graph of collocations between all the tokens in the text. In this implementation we use only unigrams, however, the usage of n-grams would clearly improve the performance. In Figure 1 this phase is represented as a graph of collocations; "E" stands for target entity.

In the **second** step we compute the PageRank of the nodes in the graph. The nodes with the highest PageRank have a thicker border in the second subfigure in Figure 1.

In the **third** step we introduce the hypernyms into the process and for each node we compute the following measure

$$m(n) := PR(n) * \Big(CO(E, n) + \frac{\sum_{h \in H} CO(h, n)}{|H|}\Big),$$

where PR stands for PageRank, H is the set of hypernyms, CO is the collocation measures. We use a variant of Dice Scores [6] as a word-association measure to grab collocations. We sort the nodes with respect to $m(n)$.

In the **fourth** step we take the first node in the sorted list of nodes and identify it as a hub. Next we build a cluster around the hub. For this purpose we take each node and assess it *involvement* in the cluster by summing the Dice Scores between itself and the hub, the direct neighbors of the hub and the hypernyms. This sum is then normalized by the sum of the Dice Scores between the node and all of its neighbors. Therefore, involvement of the node is a number between 0 and 1. In contrast to the original HyperLex method, in HyperHyperLex the nodes belong to the clusters to some degree.

After the first cluster is induced we return to the sorted list and take the next node. If the node is not yet involved in other clusters (i.e. the involvement does not surpass a certain threshold) we take it as the next hub. When building the cluster around this hub, we also take only nodes whose involvement in other clusters does not surpass the threshold. When this process selects as hub a node that is not in the thesaurus, the only difference in the above procedure is that its hypernyms are not taken into account when defining the involvement of the nodes in its cluster.

After this process, each node has a score denoting its membership in a cluster. This score is used in the next step for classifying the occurrences of the target entity. The score is defined by

$$s(n) := PR(n) * I(n),$$

where I stands for involvement of the node in said cluster.

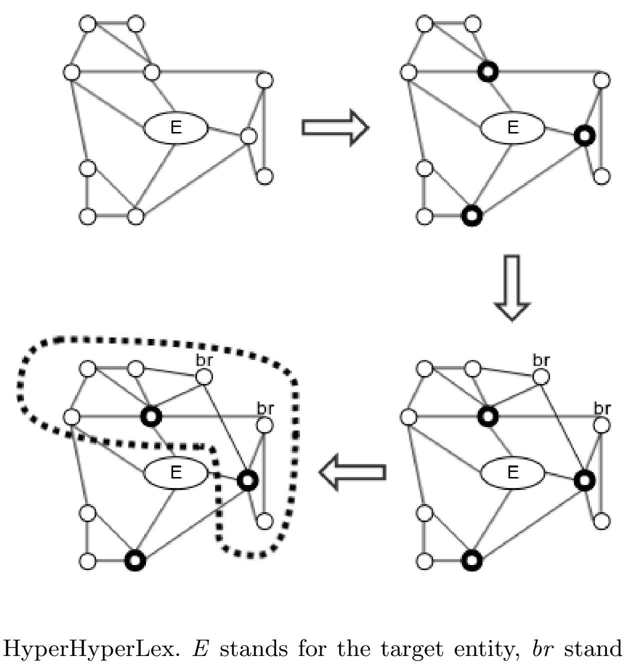

Fig. 1. HyperHyperLex. *E* stands for the target entity, *br* stand for broaders. Thick nodes identify the hubs, dashed line marks the hypernym-induced sense.

Example 2 (Americano). After calculating the collocation graph and sorting according to PageRank we get the following sorting for potential hubs

1. "campari",
2. "mix",
3. "espresso".

However, after taking hypernyms into account the highest m score is obtained by "mix". Therefore, the hypernym-induced hub is "mix", "campari" participates in this sense as it is one of the collocations of the hub. Therefore, after resorting "espresso" gets highest score and becomes the first non-hypernym-induced hub and the second overall.

In the preliminary tests we found that several senses corresponding to the target sense could be induced. However, none of the senses captured the thesaural sense completely, resulting in misclassification. With the help of the hypernyms it has become possible to capture the thesaural sense in a single sense and take into account the decisions of the data architect. In Figure 2 the sense induced with hypernyms is denoted as a dashed circle vs two solid circles induced without hypernyms. The hypernym-induced sense contains more words than the thesaural sense. Yet, the intersection of the thesaural and hypernym-induced senses is larger than that of the thesaural and non-hypernym-induced senses. This decreases the classification error.

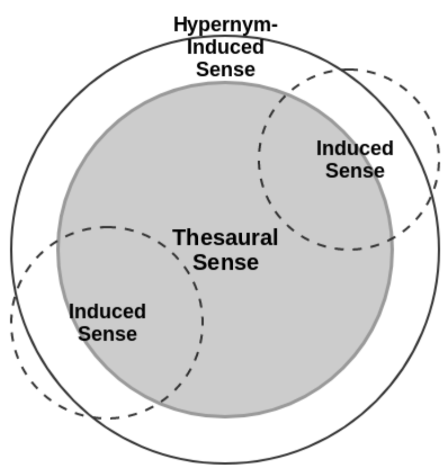

Fig. 2. WSI with and without hypernyms. The gray area indicates the "real" thesaural sense, the dashed lines mark the two sense induced with HyperLex, the solid line marks the sense induced with HyperHyperLex. Though the hypernym-induced sense may capture several general senses, it better corresponds to the thesaural sense.

3.2 WSD

The result of the previous WSI step is a set of key-value dictionaries, where

- each dictionary corresponds to an induced sense,
- the keys are the words that belong to the sense cluster,
- the values are the scores $s(n)$ or the weights of the words with respect to the sense.

In order to disambiguate an occurrence of the target word we first extract its context. A context is a set of words surrounding the target word, for example, 10 words before and 10 words after the target. Then compute the sum of the words' scores in the context with respect to each sense. For each sense we take the corresponding dictionary and sum up all the scores of all the words that are found in the context and in the sense cluster. Finally, we choose the sense with the highest aggregate score.

4 Experiment

We conduct two separate experiments with very similar setups: cocktails and MeSH [14].

4.1 Data

All the data (corpora and thesauri) can be found at `github.com/artreven/thesaural_wsi`. The two examples correspond quite well to those found in industrial settings in terms of size and depth of thesaurus, quantity, size, and quality of texts.

Thesauri In the cocktails example the concepts are taken from the "All about cocktails"[2] thesaurus. The thesaurus contains various cocktail instances and ingredients. We have only used ambiguous cocktail names for this experiment. We use the concept scheme name "cocktail" and the broaders of the target concept as hypernyms.

In the MeSH example we use the MeSH[3] thesaurus. We use the preferred labels of the broader concepts as hypernyms.

As we only use unigrams in our code, we split the compound labels into unigrams and use only nouns in both experiments.

Corpora We used the Wikilinks dataset [20] to extract the corpora. The dataset contains documents and the links to the Wikipedia pages inside the documents. The texts contain mistakes, which makes them particularly suitable for simulating a real world use case. The corpora contain duplicates. The data is used as is, without any cleaning.

First we identify those cocktail names that are ambiguous and then we collected the texts that mention those names. We remove all the texts that refer to the

[2] `vocabulary.semantic-web.at/cocktails`
[3] `www.nlm.nih.gov/mesh/`

disambiguation page at Wikipedia and all the categories containing less than 5 documents. We do the same procedure for finding ambiguous labels from MeSH.

The preprocessing phase includes removing stopwords, stemming, removing words that appear less than 3 times, removing the words that appear in more than half of the documents, and substituting all digits to a special token.

We collect 13 corpora with a total of 1227 texts for cocktails, see Table 1 for more details. We collect 8 corpora with a total of 784 texts for MeSH labels, see Table 3 for more details.

4.2 Results

In the experiment we classify the word occurrences into two categories "this" and "other". We should note that many considered words have more than 2 meanings, all the non-thesaural meanings fall into the category "other". For the baseline everything is classified into the most popular category. This baseline is known to be challenging. In practice there is no guarantee that the thesaural sense would be the most popular, therefore this baseline is better than the results one could expect in practice without WSID.

The results for cocktails are presented in Table 1. Observations:

- As can be seen from results even the high number of the real senses does not prevent the method from showing high accuracy.
- With large corpora the accuracy is high due to better WSI.
- Even if the number of thesaural mentions is very low ("Cosmopolitan", "B-52") the accuracy remains high. We assume that the well represented "other" senses can be induced accurately and the use of hypernyms improves the induction of the thesaural sense.
- The worst results are obtained for the corpora where the thesaural sense (cocktail) is the dominant: Vesper, Margarita, Martini. Since the other sense(s) are underrepresented there is a risk of getting several senses capturing the individual context. In such senses many general-purpose (not sense-specific) collocations may be present, as a result these senses may score higher in general contexts.

The results for MeSH are presented in Table 1. Observations:

- For "Warts" only one category is induced. This result is due to underrepresented "other" sense (5 documents). This is another risk for underrepresented senses.
- The MeSH corpora contains many duplicates and dirty texts. For example, sometimes the target words could be found as part of link strings or as the name of a button. Cleaning would definitely improve the results. However, even in the case of dirty data the method still yields acceptable results.

The averaged results are presented in Table 2 and in Table 4. In both cases HyperHyperLex outperforms the challenging baseline. The difference is even larger when the individual texts are taken into account (micro average), because the method benefits from having a larger corpus.

Table 1. Results of discriminating cocktail names using hypernyms

Cocktail name	Number of senses	Number of texts	Cocktail sense texts	Baseline accuracy	HyperLex	HyperHyperLex
Americano	2	45	13	0.711	0.844	0.889
Aviation	2	27	17	0.63	0.37	0.852
B-52	2	111	8	0.928	0.946	0.982
Bellini	3	95	42	0.558	0.737	0.968
Bloody Mary	6	352	109	0.69	0.827	0.935
Cosmopolitan	3	125	10	0.92	0.928	0.952
Grasshopper	4	39	13	0.667	0.744	0.974
Manhattan	4	262	70	0.733	0.851	0.885
Margarita	2	41	36	0.878	0.878	0.561
Martini	2	38	27	0.711	0.711	0.711
Mimosa	3	45	18	0.6	0.4	0.733
Tequila Sunrise	2	16	10	0.625	0.938	0.812
Vesper	2	31	24	0.774	0.839	0.677

Table 2. Averaged results for cocktails

Method	Macro average	Micro average
Baseline	0.725	0.737
HyperLex	0.784	0.821
HyperHyperLex	0.841	0.896

5 Conclusion

We have introduced a method to automatically discriminate between thesaural and non-thesaural usage of entities. The method does not require any senses of the target entity to be provided in advance and does not make use of external resources except for thesaurus, which is considered an input parameter.

In the two experiments the new method has outperformed the baseline and has shown an accuracy of about 0.9 and 0.75, respectively. The method is robust and performs well even in the real-world case of dirty data.

Table 3. Results of discriminating MeSH labels using hypernyms

MeSH label	Number of senses	Number of texts	MeSH sense texts	Baseline accuracy	HyperLex	HyperHyperLex
Amnesia	2	31	18	0.581	0.806	0.806
Dengue	2	84	59	0.702	0.298	0.583
Warts	2	14	9	0.643	0.643	0.643
Delirium	2	135	98	0.726	0.711	0.556
Iris	5	150	29	0.807	0.433	0.720
Kuru	2	69	35	0.507	0.696	0.957
Amygdala	2	64	42	0.656	0.578	0.656
Vertigo	5	237	43	0.819	0.768	0.865

Table 4. Averaged results for MeSH

Method	Macro average	Micro average
Baseline	0.68	0.735
HyperLex	0.617	0.621
HyperHyperLex	0.723	0.739

Remarks:

- We performed WSI using HyperLex for comparison. The results for some words are comparable or even better, however for other words the accuracy drops significantly. In most cases this was due to the fact shown in Figure 2, namely, there were several senses that would contain a mixture of thesaural and other sense.
- The method would benefit from using bi-grams. Indeed, all the texts contain many significant compound named entities.
- The method would benefit from using additional NLP features such as part of speech or dependencies. This would be especially useful when working with small corpora.

Acknowledgements We would like to thank Noam Ordan for thoughtful comments, careful proofreading, and valuable suggestions.

The work is partially supported by the LDL4HELTA (`ldl4.com`) project (EUREKA funding program, project ID 9898).

References

1. Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84, 2014.
2. Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 33–41, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
3. Sean Bechhofer and Alistair Miles. Skos simple knowledge organization system reference. *W3C recommendation, W3C*, 2009.
4. Timo Borst and Joachim Neubert. Case study: Publishing stw thesaurus for economics as linked open data. *W3C Semantic Web Use Cases and Case Studies*, 2009.
5. Antonio Di Marco and Roberto Navigli. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(3):709–754, 2013.
6. Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
7. Beate Dorow and Dominic Widdows. Discovering corpus-specific word senses. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, pages 79–82. Association for Computational Linguistics, 2003.

8. Ioannis P Klapaftis and Suresh Manandhar. Word sense induction using graphs of collocations. In *ECAI*, pages 298–302, 2008.
9. Jiwei Li and Dan Jurafsky. Do multi-sense embeddings improve natural language understanding? *arXiv preprint arXiv:1506.01070*, 2015.
10. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
11. Roberto Navigli. A quick tour of word sense disambiguation, induction and related approaches. *SOFSEM 2012: Theory and practice of computer science*, pages 115–129, 2012.
12. Roberto Navigli, Stefano Faralli, Aitor Soroa, Oier de Lacalle, and Eneko Agirre. Two birds with one stone: learning semantic models for text categorization and word sense disambiguation. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2317–2320. ACM, 2011.
13. Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*, 2015.
14. S. J. Nelson. Medical terminologies that work: The example of mesh. In *2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, pages 380–384, Dec 2009.
15. Julia Neuschmid, Sabrina Kirrane, Elmar Kiesling, and Thomas Thurner. *Propelling the Potential of Enterprise Linked Data in Austria.* monochrom, 2016.
16. Alexander Panchenko, Eugen Ruppert, Stefano Faralli, Simone Paolo Ponzetto, and Chris Biemann. Unsupervised does not mean uninterpretable: the case for word sense induction and disambiguation. Association for Computational Linguistics, 2017.
17. Delip Rao, Paul McNamee, and Mark Dredze. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, multilingual information extraction and summarization*, pages 93–115. Springer, 2013.
18. Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics, 2011.
19. Jennifer Rodd, Gareth Gaskell, and William Marslen-Wilson. Making sense of semantic ambiguity: Semantic competition in lexical access. *Journal of Memory and Language*, 46(2):245 – 266, 2002.
20. Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015, University of Massachusetts, Amherst, 2012.
21. Jean Véronis. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223–252, 2004.
22. David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics, 1995.
23. Zhi Zhong and Hwee Tou Ng. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83. Association for Computational Linguistics, 2010.