DAEBAK!: Peripheral Diversity for Multilingual Word Sense Disambiguation

Steve L. Manion
University of Canterbury
Christchurch, New Zealand
steve.manion
@pg.canterbury.ac.nz

Raazesh Sainudiin
University of Canterbury
Christchurch, New Zealand
r.sainudiin
@math.canterbury.ac.nz

Abstract

We introduce Peripheral Diversity (PD) as a knowledge-based approach to achieve multilingual Word Sense Disambiguation (WSD). PD exploits the frequency and diverse use of word senses in semantic subgraphs derived from larger sense inventories such as BabelNet, Wikipedia, and WordNet in order to achieve WSD. PD's f-measure scores for SemEval 2013 Task 12 outperform the Most Frequent Sense (MFS) baseline for two of the five languages: English, French, German, Italian, and Spanish. Despite PD remaining under-developed and under-explored, it demonstrates that it is robust, competitive, and encourages development.

1 Introduction

By reading out aloud "A *minute* is a *minute* division of time" (Nelson, 1976), we can easily make the distinction between the two *senses* of the homograph *minute*. For a machine this is a complex task known as Word Sense Disambiguation (WSD). Task 12 of SemEval 2013 (Navigli et al., 2013) calls for a language-independent solution to WSD that utilises a multilingual sense inventory.

Supervised approaches to WSD have dominated for some time now (Màrquez et al., 2007). Homographs such as *minute* are effortlessly disambiguated and more polysemous words such as *bar* or *line* can also be disambiguated with reasonable competence (Agirre and Edmonds, 2007). However our approach is purely knowledge-based and employs semantic graphs. This allows us to avoid the notorious predicament Gale et al. (1992) name the *information bottleneck*, in which supervised approaches fail to be portable across alternative languages and domains if the annotated corpora do not exist. Conversely, knowledge-based approaches for WSD are usually applicable to all words in unrestricted text (Mihalcea, 2007). It is this innate scalability that motivates us to pursue knowledge-based approaches. Regardless of whether sense inventories can maintain *knowledge-richness* as they grow, their continued refinement by contributors is directly beneficial.

Knowledge-based approaches that employ semantic graphs increasingly rival leading supervised approaches to WSD. They can beat a Random or LESK (Lesk, 1986) baseline (*see* Mihalcea (2005), Navigli and Lapata (2007), Sinha and Mihalcea (2007), Navigli and Lapata (2010)) and can compete with or even beat the Most Frequent Sense (MFS) baseline in certain contexts which is by no means an easy task (*see* Navigli et al. (2007), Eneko Agirre and Aitor Soroa (2009), Navigli and Ponzetto (2012a)).

2 Methodology

PD is a framework for knowledge-based WSD approaches that employ semantic graphs. However before we can elaborate we must first cover the fundamental resources it is built upon.

2.1 Fundamental Resource Definitions

2.1.1 Lemma Sequences

At a glance across the text of any language, we absorb meaning and new information through its *lexical composition*. Depending on the length of text

we are reading, we could interpret it as one of many structural subsequences of writing such as a *paragraph*, *excerpt*, *quote*, *verse*, *sentence*, among many others. Let $\mathcal{W} = (w_a, ..., w_b)$ be this subsequence of words, which we will utilise as a sliding window for PD. Again let $\mathbb{W} = (w_1, ..., w_m)$ be the larger body of text of length m, such as a *book*, *newspaper*, or *corpus of text*, that our sliding window of length $b-a$ moves through.

In SemEval Task 12 on Multilingual Word Sense Disambiguation all words are *lemmatised*, which is the process of unifying the different inflected forms of a word so they can be analysed as a consolidated *lemma* (or *headword*). Therefore words (or *lexemes*) such as *runs* and *ran* are all mapped to their unifying lemma *run*[1].

To express this, let $\ell_w : \mathcal{W} \rightarrow \mathcal{L}$ be a *many-to-one* mapping from the sequence of words $\mathcal{W}$ to the sequence of lemmas $\mathcal{L}$, in which $(w_a, ..., w_b) \mapsto (\ell_{w_a}, ..., \ell_{w_b}) = (\ell_a, ..., \ell_b)$. To give an example from the test data set[2], the word sequence $\mathcal{W}$ = (*And, it, 's, nothing, that, runs, afoul, of, ethics, rules,* .) maps to the lemma sequence $\mathcal{L}$ = (*and, it, be, nothing, that, run, afoul, of, ethic, rule,* .). In order to complete this SemEval task we disambiguate a large sequence of lemmas $\mathbb{L} = (\ell_1, ..., \ell_m)$, via our lemma-based sliding window $\mathcal{L} = (\ell_a, ..., \ell_b)$.

2.1.2 Synsets

Each lemma $\ell_i \in \mathcal{L}$ may refer up to k senses in $S(\ell_i) = \{s_{i,1}, s_{i,2}, ..., s_{i,k}\} = \mathcal{S}$. Furthermore each sense $s_{i,j} \in \mathcal{S}$ maps to a set of unique concepts in the human lexicon. To clarify let us consider one of the earliest examples of modern ambiguity taken from Bar-Hillel's (1960) critique of Machine Translation: $\mathcal{W}$ = (*The, box, was, in, the, pen,* .). The sense of *pen* could be either *a)* a certain writing *utensil* or *b)* an *enclosure* where small children can play, therefore $\{s_{\text{enclosure}}, s_{\text{utensil}}\} \subset S(\ell_{\text{pen}}) = \mathcal{S}$. Humans can easily resolve the ambiguity between the possible senses of *pen* by accessing their own internal lexicon and knowledge of the world they have built up over time.

In the same vein, when accessing sense inventories such as BabelNet, WordNet (Fellbaum, 1998), and Wikipedia which are discrete representations of the human lexicon, we refer to each sense $s_{i,j} \in \mathcal{S}$ as a synset. Depending on the sense inventory the synset belongs to, it may contain alternative or translated lexicalisations, glosses, links to other semantic resources, among a collection of semantically defined relations to other synsets.

2.1.3 Subgraphs

PD makes use of subgraphs derived from a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that can be crafted from a sense inventory, such as BabelNet, WordNet, or Wikipedia. We construct subgraphs using the BabelNet API which accesses BabelNet[3] and Babel synset paths[4] indexed into Apache Lucene[5] to ensure speed of subgraph construction. This process is described in Navigli and Ponzetto (2012a) and demonstrated in Navigli and Ponzetto (2012b). Our formalisation of subgraphs is adapted into our own notation from the original papers of Navigli and Lapata (2007) and Navigli and Lapata (2010). We refer the reader to these listed sources if they desire an extensive explanation of our subgraph construction as we have built PD on top of the same code base therefore we do not deviate from it.

For a given lemma sequence $\mathcal{L} = (\ell_i, ..., \ell_n)$ and directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we construct our subgraph $\mathcal{G}_\mathcal{L} = (\mathcal{V}_\mathcal{L}, \mathcal{E}_\mathcal{L})$ in two steps:

1. Initialize $\mathcal{V}_\mathcal{L} := \bigcup_{i=1}^{n} S(\ell_i)$ and $\mathcal{E}_\mathcal{L} := \emptyset$.
2. For each node $v \in \mathcal{V}_\mathcal{L}$, we perform a depth-first search (DFS) of $\mathcal{G}$, such that, every time we encounter a node $v' \in \mathcal{V}_\mathcal{L}$ $(v' \neq v)$ along a path $v, v_1, ..., v_k, v'$ of length $\leq L$ in $\mathcal{G}$, we add all intermediate nodes and edges on the path from v to v', i.e., $\mathcal{V}_\mathcal{L} := \mathcal{V}_\mathcal{L} \cup \{v_1, ..., v_k\}$ and $\mathcal{E}_\mathcal{L} := \mathcal{E}_\mathcal{L} \cup \{\{v, v_1\}, ..., \{v_k, v'\}\}$.

2.2 Interpretation of Problem

For the lemmatisation of any word $w_i \mapsto \ell_i : w_i \in \mathcal{W}, \ell_i \in \mathcal{L}$, we must estimate the most appropriate synset $s_{i,*} \in S(\ell_i) = \{s_{i,1}, s_{i,2}, ..., s_{i,k}\}$. Our system associates a PD score $\phi(s_{i,j})$ for each

[1] While all words are lemmatised, this task strictly focuses on the WSD of noun phrases.

[2] This is sentence `d010.s014` in the English test data set.

[3] BabelNet 1.1.1 API & Sense Inventory - `http://lcl.uniroma1.it/babelnet/download.jsp`

[4] BabelNet 1.0.1 Paths - `http://lcl.uniroma1.it/babelnet/data/babelnet_paths.tar.bz2`

[5] Apache Lucene - `http://lucene.apache.org`

$s_{i,j} \in S(\ell_i)$ by taking $\mathcal{G}_\mathcal{L}$ as input. We estimate $s_{i,*}$, the most appropriate sense for ℓ_i, by $\hat{s}_{i,*} = \arg\max_{s_{i,j} \in S(\ell_i)} \phi(s_{i,j})$. It's worth noting here that $\mathcal{G}_\mathcal{L}$ ensures the estimation of $\hat{s}_{i,*}$ is not an independent scoring rule, since $\mathcal{G}_\mathcal{L}$ embodies the context surrounding ℓ_i via our sliding lemma-based window $\mathcal{L}$.

2.3 Peripheral Diversity Framework

PD is built on the following two ideas that are explained in the following subsections:

1. For a subgraph derived from one lone lemma ℓ_i, in which no other lemmas can provide context, the synset $s_{i,j} \in \mathcal{G}_{\ell_i}$ that has the largest and most semantically diverse set of peripheral synset nodes is assumed to be the MFS for ℓ_i.

2. For a larger subgraph derived from a sliding lemma window $\mathcal{L}$, in which other lemmas can provide context, the synset $s_{i,j} \in \mathcal{G}_\mathcal{L}$ that observes the largest increase in size and semantic diversity of its peripheral synset nodes is estimated to be $s_{i,*}$, the most appropriate synset for lemma ℓ_i.

Therefore PD is merely a framework that exploits these two assumptions. Now we will go through the process of estimating $s_{i,*}$ for a given lemma ℓ_i.

2.3.1 Pairwise Semantic Dissimilarity

First, for each synset $s_{i,j} \in \mathcal{S}$, we need to acquire a set of its peripheral synsets. We do this by travelling a depth of up to d (stopping if the path ends), then adding the synset we reach to our set of peripheral synsets $\mathcal{P}^{\leq d} = \{s_{j,1}, s_{j,2}, ..., s_{j,k'}\}$.

Next for every pair of synsets v and v' that are not direct neighbours in $\mathcal{P}^{\leq d}$ such that $v \neq v'$, we calculate their Pairwise Semantic Dissimilarity (PSD) $\delta(v, v')$ which we require for a synset's PD score. To generate our results for this task we have used the complement to Cosine Similarity, commonly known as the Cosine Distance as our PSD measure:

$$\delta(v, v') = \begin{cases} 1 - \left(\frac{|O(v) \cap O(v')|}{\sqrt{|O(v)|}\sqrt{|O(v')|}} \right), & \text{if } |O(v)||O(v')| \neq 0 \\ 1, & \text{otherwise,} \end{cases}$$

where $O(v)$ is the outgoing (out-neighbouring) synsets for $v \in \mathcal{P}^{\leq d}$, and $|O(v)|$ denotes the number of elements in $O(v)$.

2.3.2 Peripheral Diversity Score

Once we have PSD scores for every permitted pairing of v and v', we have a number of ways to generate our $\phi(s_{i,j})$ values. To generate our results for this task, we chose to score synsets on the *sum of their minimum PSD values*, which is expressed formally below:

$$\phi(s_{i,j}) = \sum_{v \in \mathcal{P}^{\leq d}(s_{i,j})} \min_{\substack{v' \neq v \\ v' \in \mathcal{P}^{\leq d}(s_{i,j})}} \delta(v, v')$$

The idea is that this summing over the peripheral synsets in $\mathcal{P}^{\leq d}(s_{i,j})$ accounts for how frequently synset $s_{i,j}$ is used, then each increment in size is weighted by a peripheral synset's minimumal PSD across all synsets in $\mathcal{P}^{\leq d}(s_{i,j})$. Therefore peripheral set size and semantic diversity are rewarded simultaneously by ϕ. To conclude, the final estimated synset sequence for a given lemma sequence $(\ell_1, ..., \ell_m)$ based on ϕ is $(\hat{s}_{1,*}, \hat{s}_{2,*}, ..., \hat{s}_{m,*})$.

2.3.3 Strategies, Parameters, & Filters

Wikipedia's *Did You Mean?* We account for deviations and errors in spelling to ensure lemmas have the best chance of being mapped to a synset. Absent synsets in subgraph $\mathcal{G}_\mathcal{L}$ will naturally degrade system output. Therefore if $\ell_i \mapsto \emptyset$, we make an HTTP call to Wikipedia's *Did you mean?* and parse the response for any alternative spellings. For example in the test data set[6] the misspelt lemma: "feu_de_la_rampe" is corrected to "*feux*_de_la_rampe".

Custom Back-off Strategy As *back-off strategies*[7] have proved useful in (Navigli and Ponzetto, 2012a) and (Navigli et al., 2007), we designed our own back-off strategy. In the event our system provides a `null` result, the Babel synset $s_{i,j} \in S(\ell_i) = \mathcal{S}$ with the most senses associated with it will be chosen with preference to its region in BabelNet such that WIKIWN $\succ$ WN $\succ$ WIKI.

[6] Found in sentence `d001.s002.t005` in the French test data set.

[7] In the event the WSD technique fails to provide an answer, a back-off strategy provides one for the system to output.

Input Parameters We set our sliding window length $(b - a)$ to encompass 5 sentences at a time, in which the step size is also 5 sentences. For subgraph construction the maximum length $L = 3$. Finally we set our peripheral search depth $d = 3$.

Filters For the purposes of reproducibility only we briefly mention two filters we apply to our subgraphs that ship with the BabelNet API. We remove WordNet contributed domain relations with the `ILLEGAL_POINTERS` filter and apply the `SENSE_SHIFTS` filter. For more information on these filters we suggest the reader consult the BabelNet API documentation.

3 Results & Discussion

3.1 Results of SemEval Submission

	Language	DAEBAK!	$\text{MFS}_{\text{Baseline}}$	+/-
DE	*German*	59.10	68.60	-9.50
EN	*English*	60.40	65.60	-5.20
ES	*Spanish*	60.00	64.40	-4.40
FR	*French*	53.80	50.10	+3.70
IT	*Italian*	61.30	57.20	+4.10
	Mean	58.92	61.18	-2.26

Table 1: DAEBAK! vs MFS Baseline on BabelNet

As can be seen in Table 1, the results of our single submission were varied and competitive. The worst result was for German in which our system fell behind the MFS baseline by a margin of 9.50. Again for French and Italian we exceeded the MFS baseline by a margin of 3.70 and 4.10 respectively. Our Daebak back-off strategy contributed anywhere between 1.12% (for French) to 2.70% (for Spanish) in our results, which means our system outputs a result without the need for a back-off strategy at least 97.30% of the time. Overall our system was slightly outperformed by the MFS baseline by a margin of 2.26. Overall PD demonstrated to be robust across a range of European languages. With these preliminary results this surely warrants further investigation of what can be achieved with PD.

3.2 Exploratory Results

The authors observed some inconsistencies in the task answer keys across different languages as Table 2 illustrates. For each Babel synset ID found in the answer key, we record where its original source synsets are from, be it Wikipedia (WIKI), WordNet (WN), or both (WIKIWN).

	Language	WIKI	WN	WIKIWN
DE	*German*	43.42%	5.02%	51.55%
EN	*English*	10.36%	32.11%	57.53%
ES	*Spanish*	30.65%	5.40%	63.94%
FR	*French*	40.81%	6.55%	52.64%
IT	*Italian*	38.80%	7.33%	53.87%

Table 2: BabelNet Answer Key Breakdown

This is not a critical observation but rather an empirical enlightenment on the varied mechanics of different languages and the amount of development/translation effort that has gone into the contributing subparts of BabelNet: Wikipedia and WordNet. The heterogeneity of hybrid sense inventories such as BabelNet creates new obstacles for WSD, as seen in (Medelyan et al., 2013) it is difficult to create a disambiguation policy in this context. Future work we would like to undertake would be to investigate the heterogenous nature of BabelNet and how this affects various WSD methods.

4 Conclusion & Future Directions

To conclude PD has demonstrated in its early stages that it can perform well and even outperform the MFS baselines in certain experimental contexts. Furthermore it leaves a lot left to be explored in terms of what this approach is capable of via adjusting subgraph filters, strategies, and input parameters across both heterogenous and homogenous semantic graphs.

Acknowledgments

This research was completed with the help of the Korean Foundation Graduate Studies Fellowship[8].

5 Resources

The code base for this work can be found in the near future at `http://www.stevemanion.com/`.

[8]KF Graduate Studies Fellowship - `http://www.kf.or.kr/eng/01_sks/sks_fel_sfb01.asp`

References

Eneko Agirre and Philip Edmonds. 2007. Introduction. *Word Sense Disambiguation Algorithms and Applications*, Chapter 1:1-28. Springer, New York.

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for Word Sense Disambiguation. *In Proceedings of the 12th Conference of the European Chapter of the ACL*, April:33–41. Association for Computational Linguistics.

Yehoshua Bar-Hillel. 1960. The Present Status of Automatic Translation of Languages. *Advances in Computers*, 1:91–163.

Christiane Fellbaum. 1998, ed. *WordNet: An Electronic Lexical Database.*, Cambridge, MA: MIT Press.

William A Gale, Kenneth W Church, David Yarowsky. 1992. A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, 26(5–6):415–439.

Michael Lesk. 1986. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. *Proceedings of the 5th Annual International Conference on System Documentation.*, 24–26. ACM.

Llus Màrquez, Gerard Escudero, David Martínez, German Rigau. 2007. Supervised Corpus-Based Methods for WSD. *Word Sense Disambiguation Algorithms and Applications*, Chapter 7:167-216. Springer, New York.

Rada Mihalcea. 2005. Unsupervised Large-Vocabulary Word Sense Disambiguation with Graph-based Algorithms for Sequence Data Labeling. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 411-418. Association for Computational Linguistics.

Rada Mihalcea. 2007. Knowledge-Based Methods for WSD. *Word Sense Disambiguation Algorithms and Applications*, Chapter 5:107–131. Springer, New York.

Alyona Medelyan, Steve Manion, Jeen Broekstra, Anna Divoli, Anna-lan Huang, and Ian H Witten. 2013. Constructing a Focused Taxonomy from a Document Collection *Extended Semantic Web Conference*, (Accepted, in press)

Roberto Navigli and Mirella Lapata. 2007. Graph connectivity measures for unsupervised word sense disambiguation. *IJCAI'07 Proceedings of the 20th International Joint Conference on Artifical Intelligence*, 1683–1688.

Roberto Navigli, Kenneth C Litkowski, and Orin Hargraves. 2007. SemEval-2007 Task 07: Coarse-Grained English All-Words Task. *In Proceedings of the 4th International Workshop on Semantic Evaluations*, 30–35.

Roberto Navigli and Mirella Lapata. 2010. An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):678–692.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Multilingual WSD with Just a Few Lines of Code: the BabelNet API. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 67–72.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantcis (*SEM 2013)*.

Frederic Nelson. 1976. Homographs *American Speech*, 51(3):296–297.

Ravi Sinha and Rada Mihalcea. 2007. Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. *Proceedings of IEEE International Conference on Semantic Computing*.