Duluth: Word Sense Discrimination in the Service of Lexicography

Ted Pedersen
Department of Computer Science
University of Minnesota
Duluth, MN, 55812, USA
tpederse@d.umn.edu

Abstract

This paper describes the Duluth systems that participated in Task 15 of SemEval 2015. The goal of the task was to automatically construct dictionary entries (via a series of three subtasks). Our systems participated in subtask 2, which involved automatically clustering the contexts in which a target word occurs into its different senses. Our results are consistent with previous word sense induction and discrimination findings, where it proves difficult to beat a baseline algorithm that assigns all instances of a target word to a single sense. However, our method of predicting the number of senses automatically fared quite well.

1 Introduction

A Corpus Pattern Analysis (CPA) dictionary entry building task (SemEval 2015 Task 15) included three subtasks, the combination of which creates a dictionary entry based on CPA (Hanks, 2013). The Duluth systems participated in the second subtask, which sought to cluster the contexts in which target words occur based on their underlying sense or meaning. Note that for this task all of the target words are verbs. This is unusual for a word sense shared task, since nouns are much more commonly studied.

The task input includes two sets of words : the Microcheck includes 8 target verbs, where the number of senses for each are given to task participants, while the Wingspread includes 20 target verbs where the number of senses are withheld. Both sets of target verbs and their frequencies are shown in Tables 3.2 and 3.2.

The CPA method is based on finding patterns of use in corpora, and definitions of word senses refer explicitly to these patterns. For example, the verb *totter* has three senses, where a person (sense 1), building (sense 2), or institution (sense 3) may be what totters. The verb *undertake* has two senses, where a person or institution embarks on an activity (sense 1) or promises to do so (sense 2).

There is certainly a role for syntactic information in defining such senses – direct and indirect objects are clearly important, and chunking would in general be quite useful. It also seems that incorporating semantic features, for example, those based on selectional restrictions or constraints, might be fruitful. In fact, subtask 1 focuses on shallow parsing and is said to be similar to semantic role labeling. Given different syntactic and semantic features discovered in subtask 1, it would be possible to pursue subtask 2 using a more rule based approach.

However, the Duluth systems do not explicitly account for syntax or semantics and do not try to identify these kinds of patterns. While we believe such approaches are extremely useful, we are primarily interested in exploring the limits of methods that depend on purely lexical features.

As a result, the Duluth systems rely on clustering target verbs based on the context in which they occur (e.g., (Schütze, 1998), (Purandare and Pedersen, 2004), (Pedersen, 2007)). This follows from the distributional hypothesis (Harris, 1954). Simply put, words that are used in similar contexts may often have similar meanings. However, words with different meanings can also be used in similar contexts (e.g., antonyms) so results are often noisy.

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 282–286,
Denver, Colorado, June 4-5, 2015.

The Duluth systems take a knowledge-lean approach (Pedersen, 1997), and treat this task as an unsupervised word sense discrimination or induction problem, and use the freely available open-source software package SenseClusters[1].

2 Systems

We submitted three runs for subtask 2 : run1, run2, and run3. These three systems share a few basic characteristics, but differ in important respects. All use SenseClusters, and all utilize the same relatively simple pre–processing. Text was converted to lower case, and numeric values were all converted to a single string. Also, all three runs automatically determined the number of clusters (senses) using the PK2 measure (Pedersen and Kulkarni, 2006). This measure looks at the degree of change in the clustering criterion function, and stops the clustering process when the criterion function begins to plateau. This indicates that additional clustering of the data is not improving the quality of the clusters, and that further divisions will break apart relatively homogeneous senses.

There are however important differences between the systems. Runs run1 and run2 rely on second–order co–occurrences, run1 uses words that co–occur near the target verb as features, and run2 uses words that occur anywhere in the contexts to be clustered. Both run1 and run2 represent these features using second–order co–occurrences, where run1 derives these from the contexts to be clustered, and run2 uses the WordNet 3.0 glosses[2] as a 1.46 million word corpus for building these features. run3 use first–order unigrams found in the contexts to be clustered as features.

While the Microcheck data provided the number of senses, the Duluth systems elected not to use this. We felt that in most realistic use cases the number of senses is not known, and we were curious to see how well our systems could perform at identifying the number of senses automatically.

2.1 First and Second–Order Co–Occurrences

A first–order representation simply looks for features that directly occur in the contexts to be clustered and uses their occurrence (or not) as the basis for making clustering decisions. First–order unigrams depend on having multiple occurrences of the same words in various different contexts, and as such often do not perform well with smaller numbers of contexts. Among our systems, run3 is the only to take a first order unigram approach.

A second–order representation takes a somewhat fuzzier approach, and allows for a more flexible sort of feature matching. Rather than looking for the same features in multiple contexts, this representation seeks features that co–occur with the same words in different contexts. This can be thought of as a kind of a *friend of a friend* approach to feature matching.

For example, suppose that *car* and *auto* occur in two different contexts. They do not match (as first–order features) but if both are known to occur with *repairs* then that second–order co–occurrence can be the basis for considering them as matching features that could then be used to cluster the contexts in which *car* and *auto* occur in together.

This is operationalized by replacing words in the context to be clustered with a co–occurrence vector. For run1, the only word that is replaced is the target verb, which is instead represented by a vector of words that occur within 8 positions of that target in that particular context.

For run2, all the words in the contexts to be clustered that are used in a WordNet gloss (version 3.0) are replaced by a vector representing all the words in WordNet glosses that immediately follow that word in a definition.

As a simple example, imagine a gloss corpus with two definitions : *a vehicle powered by an internal combustion engine* and *a medication used to speed up the internal clock*. If the word *internal* occurs in a context, it would be replaced by a vector consisting of *combustion* and *clock*.

Then, all the vectors associated with the words in a context are averaged together (although in the case of run1 this might just be a single vector). Each context is represented now by its averaged vector, and the closeness or distance of contexts to or from each other is based on the number of second–order feature matches.

[1] http://senseclusters.sourceforge.net

[2] http://www.d.umn.edu/~tpederse/Code/glossExtract-v0.03.tar.gz

	Microcheck	Wingspread
run1	0.525	0.604
run2	0.440	0.581
run3	0.439	0.615
baseline	0.588	0.720

Table 1: B–Cubed F–Scores.

2.2 Lexical Feature Selection

run1 finds what are known in SenseClusters as target co–occurrences (tco) in the contexts to be clustered, and run2 finds bigrams in the WordNet 3.0 gloss corpus. While there are many methods for identifying statistically significant or associated pairs of words in corpora, the number of contexts in the Wingspread data is relatively small – 12 of 20 target verbs have fewer than 40 contexts, so we simply relied on frequency counts when selecting features. Given this, run1 used a long distance definition of co–occurrence to help overcome the smaller numbers of contexts, and so any word that occurs anywhere within 8 positions of the target word 2 or more times is considered a target co–occurrence. In run2 any bigram that occurred 5 or more times in the WordNet 3.0 gloss corpus was used as a feature. In run3 any unigram that occurred 2 or more times in the contexts to be clustered was used as a feature.

We used the nearly 400 word stoplist from the Ngram Statistics Package[3] (Banerjee and Pedersen, 2003) for all three of our runs. Any bigram or co–occurrence where both words are stop words was not used as a feature, and any unigram in the stoplist was likewise discarded.

3 Results and Analysis

Official results from task 15 are based on the B–cubed F–score (Bagga and Baldwin, 1998). In addition to reporting those values, we also carried out our own analysis using the SenseClusters F–measure.

3.1 B–cubed F–score

Table 3.1 shows the B–Cubed F–scores as reported by the task organizers. Note that the baseline system assigns all contexts to a single cluster or sense.

Prior to the evaluation we designated run1 as our official submission, since we felt that this system was likely to be most successful with this task. This was based on our pre–evaluation tuning with the training data which had been made available by the task organizers. This prediction was largely confirmed – run1 was easily our most accurate system with the Microcheck data, and was only narrowly exceeded by run3 for the Wingspread data.

There were several hundred contexts available for each target verb in the Microcheck data. This is large enough to generate a rich second–order representation of context. Given that we focused on somewhat localized target co–occurrences in run1, the number of spurious features will be somewhat less than if we had looked more generally at features that occur anywhere in a context (as is the case with run2 and run3). This is why we believe that run1 had a fairly significant advantage in the Microcheck data.

However, in the Wingspread data run3 slightly outperformed run1, although not to a significant degree. We believe this occurred because the Wingspread data has a majority of target verbs with less than 40 contexts. This small amount of data will result in very sparse second–order co–occurrences. Given that run1 seeks target co–occurrences, when these are very sparse they essentially reduce to first–order co–occurrences, leading to very similar performance between run1 and run3.

3.2 SenseClusters F–Measure

Tables 3.2 and 3.2 provide results for run1 using the SenseClusters F-Measure (F) (Pedersen, 2007). This measure first assigns the discovered clusters to gold standard senses in whatever way optimizes the agreement between them using the (Munkres, 1957) algorithm. Then any senses or clusters that are not aligned are discarded, and precision and recall are computed in the usual way. In these experiments all contexts are assigned to clusters, so recall and precision are the same, and the F-measure can be viewed as accuracy. In this case the F-measure is the percentage of contexts that were assigned to the correct cluster.

These tables also show the most frequent sense baseline (M). This is the percentage of contexts that belong to the most frequent sense. This is a well known baseline in supervised approaches to word sense disambiguation, and also proves to be the same for unsupervised approaches. Given the defini-

[3] http://ngram.sourceforge.net

	N	C	D	M	F
appreciate	215	2	2	.744	.693
apprehend	123	3	5	.626	.435
continue	203	7	4	.350	.291
crush	170	5	5	.365	.324
decline	201	3	4	.672	.439
operate	140	8	4	.286	.250
undertake	228	2	2	.895	.750
total (w)		4.1	3.5	.585	.478
total	1,280	4.3	3.7	.562	.455

Table 2: Microcheck run1, N is number of instances, C is number of actual clusters, D is number of discovered clusters, M is majority sense baseline, F is SenseClusters F-Measure, total (w) are weighted averages.

tion of the SenseClusters F-Measure, if all contexts are assigned to a single cluster, then the F-Measure will be equal to the most frequent sense percentage. As can be seen in Tables 2 and 3, in general this baseline outperformed the Duluth systems for nearly every target verb.

We were pleased that in general the PK2 method of identifying the number of clusters was reasonably successful. While it did not always predict exactly the same number of clusters as found in the gold standard data, in general there were no cases where it differed radically. On average the Microcheck data had 4.3 senses, while run1 discovered 3.7. For the Wingspread data there were 3.0 senses, while run1 discovered 2.7. While the results show that the clusters themselves are noisy, in general we are pleased that our ability to predict the number of clusters is reasonably accurate.

4 Conclusions

SenseClusters has participated in numerous SensEval and SemEval shared tasks that have included word sense discrimination and induction (Pedersen, 2007; Pedersen, 2010; Pedersen, 2013). In all of these prior events, the most frequent sense baseline has proven hard to beat. In general assigning all instances of a target verb to a single cluster replicates most frequent sense performance. The results in this subtask are similar, and suggest that for the moment, automatic word sense discrimination is still not a viable replacement for human lexicographic expertise.

	N	C	D	M	F
adapt	182	4	1	.539	.539
advise	230	8	2	.365	.365
afflict	179	2	2	.961	.687
ascertain	7	2	1	.571	.571
ask	573	9	2	.522	.470
attain	240	3	4	.833	.627
avert	240	2	7	.958	.374
avoid	242	3	2	.727	.566
begrudge	19	2	4	.579	.581
belch	24	3	4	.583	.468
bludgeon	32	2	2	.500	.500
bluff	25	2	2	.560	.520
boo	36	2	2	.750	.640
brag	29	2	2	.621	.586
breeze	12	2	1	.583	.583
sue	247	2	2	.980	.846
teeter	28	2	2	.821	.750
tense	37	3	2	.622	.432
totter	19	2	5	.632	.533
wing	22	2	4	.474	.864
total (w)		4.6	2.7	.694	.548
total	2,421	3.0	2.7	.659	.575

Table 3: Wingspread run1, N is number of instances, C is number of actual clusters, D is number of discovered clusters, M is majority sense baseline, F is SenseClusters F-Measure, total (w) are weighted averages.

However, we are encouraged by the accurate results from the PK2 method in identifying the number of senses automatically. If the discovered clusters themselves can be made less noisy (through improved feature selection), our overall results could improve significantly since we are already able to identify the number of distinct senses accurately. We believe that the incorporation of more grammatical and semantic features will certainly help improve the quality of the clustering, and so plan to pursue that in future work.

Acknowledgments

I would like to thank Bridget McInnes for her help in understanding the task, and for very useful brainstorming discussions.

References

Amit Bagga and Breck Baldwin. 1998. Entity–based cross–document co–referencing using the vector space model. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 79–85.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810, Acapulco, August.

Patrick Hanks. 2013. *Lexical Analysis : Norms and Exploitations*. The MIT Press, Cambridge, MA.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

James Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, March.

Ted Pedersen and Anagha Kulkarni. 2006. Selecting the right number of senses based on clustering criterion functions. In *Proceedings of the Posters and Demo Program of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics*, pages 111–114, Trento, Italy, April.

Ted Pedersen. 1997. Knowledge lean word sense disambiguation. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, page 814, Providence, RI, July.

Ted Pedersen. 2007. UMND2 : SenseClusters applied to the sense induction task of Senseval-4. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 394–397, Prague, Czech Republic, June.

Ted Pedersen. 2010. Duluth-WSI: SenseClusters applied to the sense induction task of semEval-2. In *Proceedings of the SemEval 2010 Workshop : the 5th International Workshop on Semantic Evaluations*, pages 363–366, Uppsala, July.

Ted Pedersen. 2013. Duluth : Word sense induction applied to web page clustering. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 202–206, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Amruta Purandare and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 41–48, Boston, MA.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.