Duluth : Word Sense Induction Applied to Web Page Clustering

Ted Pedersen
Department of Computer Science
University of Minnesota
Duluth, MN 55812 USA
tpederse@d.umn.edu
http://senseclusters.sourceforge.net

Abstract

The Duluth systems that participated in task 11 of SemEval–2013 carried out word sense induction (WSI) in order to cluster Web search results. They relied on an approach that represented Web snippets using second–order co–occurrences. These systems were all implemented using SenseClusters, a freely available open source software package.

1 Introduction

The goal of task 11 of SemEval–2013 was to cluster Web search results (Navigli and Vannella, 2013). The test data consisted of the top 64 Google results for each of 100 potentially ambiguous queries, for a total of 6,400 test instances. The Web snippets returned for each query were clustered and evaluated separately, with an overall evaluation score provided for each system.

The problem of Web page clustering is one of the use cases envisioned for SenseClusters (Pedersen and Kulkarni, 2007; Pedersen, 2010a), a freely available open source software package developed at the University of Minnesota, Duluth starting in 2002. It supports first and second–order clustering of contexts using both co–occurrence matrices (Purandare and Pedersen, 2004; Kulkarni and Pedersen, 2005) and Latent Semantic Analysis (Landauer and Dumais, 1997).

SenseClusters has participated in various forms at different SenseEval and SemEval shared tasks, including SemEval-2007 (Pedersen, 2007), SemEval-2010 (Pedersen, 2010b) and also in an i2b2 clinical medicine task (Pedersen, 2006).

2 Duluth System

While we refer to three Duluth systems (sys1, sys7, and sys9), in reality these are all variations of the same overall system. All three are based on second–order context clustering as provided in SenseClusters. The query terms are treated exactly like any other word in the snippets, which is called *headless* clustering in SenseClusters.

2.1 Common aspects to all systems

The input to sys1, sys7, and sys9 consists of 64 Web search snippets, each approximately 25 words in length. All text was converted to upper case prior to processing. The goal was to group the 64 snippets for each query into k distinct clusters, where k was automatically determined by the PK2 method of SenseClusters (Pedersen and Kulkarni, 2006a; Pedersen and Kulkarni, 2006b). Each discovered cluster represents a different underlying meaning of the given query term that resulted in those snippets being returned. Word sense induction was carried out separately on the Web snippets associated with each query term, meaning that the algorithm was run 100 times and clustered 64 Web page snippets each time.

In second–order context clustering, the words in a context (i.e., Web snippet) to be clustered are replaced by vectors that are derived from some corpus of text. The corpora used are among the main differences in the Duluth systems. Once the words in a context are replaced by vectors, those vectors are averaged together to create a new representation of the context. That representation is said to be *second–order* because each word is represented by its direct or first order co–occurrences, and simi-

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 202–206, Atlanta, Georgia, June 14-15, 2013. ©2013 Association for Computational Linguistics

larities between words in the same Web snippet are captured by the set of words that mutually co–occur with them.

If *car* is represented by the vector *[motor, magazine, insurance]*, and if *life* is represented by the vector *[sentence, force, insurance]*, then *car* and *life* are said to be second–order co-occurrences because they both occur with *insurance*. A second–order co-occurrence can capture more indirect relationships between words, and so these second–order connections tend to be more numerous and more subtle than first–order co–occurrences (which would require that *car* and *life* co–occur near or adjacent to each other in a Web snippet to establish a relationship).

The co-occurrence matrix is created by finding bigrams that occur more than a given number of times (this varies per system) and have a log-likelihood ratio greater than 3.84.[1] Then, the first word in a bigram is represented in the rows of the matrix, the second word is represented in the columns. The value in the corresponding cell is the log-likelihood score. This matrix is therefore not symmetric, and has different entries for *old age* and *age old*. Also, any bigram that includes one or two stop words (e.g., *to fire, running to, for the*) will be excluded and not included in the co-occurrence matrix and will not be included in the overall sample count used for computing the log–likelihood ratio. To summarize then, words in a Web snippet are represented by the words with which they occur in bigrams, where the context word is the first word in the bigram, and the vector is the set of words that follow it in bigrams.

Once the co–occurrence matrix is created, it may be optionally reduced by Singular Value Decomposition. The result of this will be a matrix with the same number of rows prior to SVD, but a reduced number of columns. The goal of SVD is to compress together columns of words with similar co–occurrence patterns, and thereby reduce the size and noisiness of the data. Whether the matrix is reduced or not, then each word in each snippet to be clustered is replaced by a vector from that matrix. A word is replaced by the row in the co-occurrence matrix to which it corresponds. Any words that do not have an entry in the co-occurrence matrix will not be represented. Then, the contexts are clustered using the method of repeated bisections (Zhao and Karypis, 2004), where the number of clusters is automatically discovered using the PK2 method.

2.2 Differences among systems

The main difference among the systems was the corpora used to create their co-occurrence matrices.

The smallest corpus was used by sys7, which simply treated the 64 snippets returned by each query as the corpus for creating a co–occurrence matrix. Thus, each query term had a unique co-occurrence matrix that was created from the Web snippets returned by that query. This results in a very small amount of data per query (approx. 25 words/snippet * 64 snippets = 1600 words), and so bigrams were allowed to have up to three intervening words that were skipped (in order to increase the number of bigrams used to create the co–occurrence matrix). Bigrams were excluded if they only occurred 1 time, had a log–likelihood ratio of less than 3.84, or were made up of one or two stop words. Even with this more flexible definition of bigram, the resulting co–occurrence matrices were still quite small. The largest resulting co–occurrence matrix for any query was 221 x 222, with 602 non–zero values (meaning there were 602 different bigrams used as features). The smallest of the co-occurrence matrices was 102 x 113 with 242 non–zero values. Given these small sizes, SVD was not employed in sys7.

sys1 and sys9 used larger corpora, and therefore required bigrams to be made up of adjacent words that occurred 5 or more times, had log–likelihood ratio scores of 3.84 or above, and contained no stop words. Rather than having a different co–occurrence matrix for each query, sys1 and sys9 created a single co-occurrence matrix for all queries.

In sys1, all the Web snippet results for all 100 queries were combined into a single corpus. Thus, the co–occurrence matrix was based on bigram features found in a corpus of 6,400 Web snippets that consisted of approximately 160,000 words. This resulted in a co–occurrence matrix of size 771 x 952 with 1,558 non–zero values prior to SVD. After SVD the matrix was 771 x 90, and all cells had non-

[1] This value corresponds with a p-value of 0.05 when testing for significance, meaning that bigrams with log-likelihood at least equal to 3.84 have at least a 95% chance of having been drawn from a population where their co-occurrence is not by chance.

zero values (as a result of SVD). Note that if there are less than 3,000 columns in a co-occurrence matrix, the columns are reduced down to 10% of their original size. If there are more than 3,000 columns then it is reduced to 300 dimensions. This follows recommendations for SVD given for Latent Semantic Analysis (Landauer and Dumais, 1997).

Rather than using task data, sys9 uses the first 10,000 paragraphs of Associated Press newswire (APW) that appear in the English Gigaword corpus (1st edition) (Graff and Cieri, 2003). This created a corpus of approximately 3.6 million words which resulted in a co-occurrence matrix prior to SVD of 9,853 x 10,995 with 43,199 non-zero values. After SVD the co–occurrence matrix was 9,853 by 300.

3 Results

Various measures were reported by the task organizers, including F1 (F1-13), the Rand Index (RI), the Adjusted Rand Index (ARI), and the Jaccard Coefficient. More details can be found in (Di Marco and Navigli, 2013).

In addition we computed the paired F-Score (F-10) (Artiles et al., 2009) as used in the 2010 SemEval word sense induction task (Manandhar et al., 2010) and the F-Measure (F-SC), which is provided by SenseClusters. This allows for the comparison of results from this task with the 2010 task and various results from SenseClusters.

The organizers also provided scores for S-recall and S-precision (Zhai et al., 2003), however for these to be meaningful the results for each cluster must be output in ranked order. The Duluth systems did not make a ranking distinction among the instances in each cluster, and so these scores are not particularly meaningful for the Duluth systems.

3.1 Comparisons to Baselines

Table 1 includes the results of the three submitted Duluth systems, plus numerous baselines. **RandX** designates a random baseline where senses were assigned by randomly assigning a value between 1 and X. In word sense induction, the labels assigned to discovered clusters are arbitrary, so a random baseline is a convenient sanity check. **MFS** replicates the most frequent sense baseline from supervised learning by simply assigning all instances for a word to a single cluster. This is sometimes also known as the "all–in–one" baseline. **Gold** are the evaluation results when the gold standard data is provided as input (and compared to itself).

The various baselines give us a sense of the characteristics of the different evaluation measures, and a few points emerge. We have argued previously (Pedersen, 2010a) that any evaluation measure used for word sense induction needs to be able to expose random baselines and distinguish them from more systematic results. By this standard a number of measures are found to be lacking. In SemEval–2010 we demonstrated that the V-Measure (Rosenberg and Hirschberg, 2007) had an overwhelming bias towards systems that produce larger numbers of clusters – as a result it scored random baselines that generated larger number of clusters (like Rand25 and Rand50) very highly.

The Rand Index (**RI**), which does not correct for chance agreement, also scores random baselines higher than both non–random systems and MFS. The Adjusted Rand Index (**ARI**) corrects for chance and scores random systems near 0, but it also scores MFS near 0. According to ARI, random systems and MFS perform at essentially the same level. This is a troublesome tendency when evaluating word sense induction systems, since MFS is often considered a reasonable baseline that provides useful results. Many words have relatively skewed distributions where they are mostly used in one sense, and this is exactly what is approximated by MFS.

Of the measures included in Table 1, the paired FScore (F-10), the F-Measure (F-SC), and the Jaccard Coefficient provide results that seem most appropriate for word sense induction. This is because these measures score random baselines lower than MFS, and that RandX scores lower than RandY, when (X $>$ Y). The paired FScore (F-10) and the Jaccard Coefficient arrived at similar results, where Rand50 received an extremely low score, and MFS scored the highest. The F-measure (F-SC) had a similar profile, except that the decline in evaluation scores as X grows in RandX is somewhat less.

The paired F-Score (F-10), the F-Measure (F-SC), and F1 (F1-13) all score MFS at approximately 54%, which is intuitively appealing since that is the percentage of instances correctly clustered if all instances are placed into a single cluster. However, in

Table 1: Experimental Results

System	F-10	F-SC	Jaccard	F1-13	RI	ARI	clusters	size
sys1	46.53	46.90	31.79	56.83	52.18	5.75	2.5	26.5
sys7	45.89	44.03	31.03	58.78	52.04	**6.78**	3.0	25.2
sys9	35.56	37.21	22.24	57.02	54.63	2.59	3.3	19.8
Rand2	41.49	42.86	26.99	54.89	50.06	-0.04	2.0	32.0
Rand5	25.17	31.28	14.52	56.73	56.13	0.12	5.0	12.8
Rand10	15.05	28.71	8.18	59.67	58.10	0.02	10.0	6.4
Rand25	7.01	26.78	3.63	66.89	59.24	-0.15	23.2	2.8
Rand50	4.07	25.97	2.00	**76.19**	**59.73**	0.10	35.9	1.8
MFS	**54.06**	**54.42**	**39.90**	54.42	39.90	0.0	1.0	64.0
Gold	100.00	100.00	100.00	100.00	100.00	99.0	7.7	11.6

other cases these measures begin to diverge. F1 (F1-13) tends to score random baselines even higher than MFS, and Rand50 gets a higher score than Rand2, which is somewhat counter intuitive. In fact according to F1 (F1-13), Rand50 would have been the top ranked system in task 11. It appears that F1 (F1-13) is strongly influenced by cluster purity, but does not penalize a system for creating too many clusters. Thus, as the number of clusters increases, F1 (F1-13) will consistently improve since smaller clusters are nearly always more pure than larger ones.

Interestingly enough, the Rand Index (RI) and the Jaccard Coefficient both score MFS at 39%. This number does not have an intuitively appealing interpretation, and thereafter RI and Jaccard diverge. RI scores random baselines higher than MFS, whereas the Jaccard Coefficient takes the more reasonable path of scoring random baselines well below MFS.

3.2 Duluth Systems Evaluation

The FScore (F-10), F-Measure (F-SC), and Jaccard Coefficient result in a comparable and consistent view of the system results. sys1 was found to be the most accurate, followed closely by sys7. All three measures showed that sys9 lagged considerably.

While all three systems relied on second–order co-occurrences, sys7 used the least amount of data, while sys9 used the most. This shows that better results can be obtained using the Web snippets to be clustered as the source of the co–occurrence data (as sys1 and sys7 did) rather than larger amounts of possibly less relevant text (as sys9 did).

Each of these systems created a roughly comparable number of clusters (on average, per query term, shown in the column labeled *clusters*). sys7 created 2.53, while sys9 created 3.01, and sys1 found 3.32. The average number of web snippets in the discovered clusters (shown in the column labeled *size*) are likewise somewhat consistent: sys1 was the largest at 26.5, sys7 had 25.2, and sys9 was the smallest with 19.8. The gold standard found an average of 7.7 queries per cluster and 11.6 snippets per cluster.

After the competition sys1 and sys9 were run without SVD. There was no significant difference in results with or without SVD. This is consistent with previous work that found SVD had relatively little impact in name discrimination experiments (Pedersen et al., 2005).

4 Conclusions

sys7 achieved the best results by using very small co-occurrence matrices of approximately one to two hundred rows and columns. While small, this data was most relevant to the task since it was made up of the Web snippets to be clustered. sys1 increased the size of the co–occurrence matrix to 771 x 96 by using all of the test data, but saw no increase in performance. sys9 used the largest corpus, which resulted in a co–occurrence matrix of 9,853 x 300, and had the poorest results of the Duluth systems.

Sixty–four instances is a small amount of data for clustering. In future we will augment each query with additional unannotated web snippets that will be discarded after clustering. Hopefully the core 64 instances that remain will be clustered more effectively given the cushion provided by the extra data.

References

J. Artiles, E. Amigó, and J. Gonzalo. 2009. The role of named entities in Web People Search. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 534–542, Singapore, August.

A. Di Marco and R. Navigli. 2013. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(4):1–46.

D. Graff and C. Cieri. 2003. English Gigaword. Linguistic Data Consortium, Philadelphia.

A. Kulkarni and T Pedersen. 2005. SenseClusters: Unsupervised discrimination and labeling of similar contexts. In *Proceedings of the Demonstration and Interactive Poster Session of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 105–108, Ann Arbor, MI, June.

T. Landauer and S. Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240.

S. Manandhar, I. Klapaftis, D. Dligach, and S. Pradhan. 2010. SemEval-2010 Task 14: Word sense induction and disambiguation. In *Proceedings of the SemEval 2010 Workshop : the 5th International Workshop on Semantic Evaluations*, Uppsala, Sweden, July.

R. Navigli and D. Vannella. 2013. Semeval-2013 task 11: Evaluating word sense induction and disambiguation within an end-user application. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM-2013)*, Atlanta, June.

T. Pedersen and A. Kulkarni. 2006a. Automatic cluster stopping with criterion functions and the gap statistic. In *Proceedings of the Demonstration Session of the Human Language Technology Conference and the Sixth Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 276–279, New York City, June.

T. Pedersen and A. Kulkarni. 2006b. 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.

T. Pedersen and A. Kulkarni. 2007. Discovering identities in web contexts with unsupervised clustering. In *Proceedings of the IJCAI-2007 Workshop on Analytics for Noisy Unstructured Text Data*, pages 23–30, Hyderabad, India, January.

T. Pedersen, A. Purandare, and A. Kulkarni. 2005. Name discrimination by clustering similar contexts. In *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 220–231, Mexico City, February.

T. Pedersen. 2006. Determining smoker status using supervised and unsupervised learning with lexical features. In *Working Notes of the i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*, Washington, DC, November.

T. 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.

T. Pedersen. 2010a. Computational approaches to measuring the similarity of short contexts. Technical report, University of Minnesota Supercomputing Institute Research Report UMSI 2010/118, October.

T. Pedersen. 2010b. 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.

A. Purandare and T. 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.

A. Rosenberg and J. Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, Prague, Czech Republic, June.

C. X. Zhai, W. Cohen, and J. Lafferty. 2003. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–17. ACM.

Y. Zhao and G. Karypis. 2004. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55:311–331.