Dictionary Definitions: The Likes and the Unlikes

Anagha Kulkarni, Jamie Callan and Maxine Eskenazi

Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave,
Pittsburgh, PA, USA 15213
{anaghak,callan,max}@cs.cmu.edu

Abstract

The task of grouping word definitions from ESL (English as a Second Language) dictionaries based on the similarity of their meanings is the focus of this work. It is demonstrated that lexical features and unsupervised machine learning algorithms can be effectively used to approach this problem. Analysis of the efficacy of this methodology for this task and the involved data which consists of very short and very few definitions per group is provided.

Index Terms: Clustering, Unsupervised machine learning, Computer Assisted Language Learning

1. Introduction

The REAP tutoring system [1] provides assistance to ESL students to improve their vocabulary. For every grade level a human teacher provides a list of words (focus-words) that the students should learn and practice in an academic semester. These words are drawn from the AWL (Academic Word List). Currently, it is this set of focus-words that REAP assists the students with. The approach is to teach the focus-words using descriptive methods in addition to prescriptive methods. This is achieved by providing reading material that demonstrates usage of one or more focus-words in real text and thus effectively implements context-based teaching. Reading material is selected from the World Wide Web (WWW). The suitability of the reading material is maintained by selecting only those documents from the WWW that stand the tests of various automatic filters implemented in the REAP system, for instance, text-quality filter, reading-level filter and document length filter. To maximize the time on task, documents containing multiple focus-words are preferred. Students can also provide topical preferences such as, Arts, Science, and Sports, which are taken into consideration by the system while choosing the documents for the student. As such, the focus is on improving student's vocabulary by showing the words in their natural neighborhood instead of in isolation and further more providing the words in student preferred context (the topic preference) and thus increasing the student motivation as well.

A more direct source of word's meaning, an ESL dictionary, a machine readable version of the Cambridge Advanced Learners Dictionary (CALD) [2] is integrated in REAP. Students can use CALD while they read a given document, to lookup the focus or non-focus words. After a student has finished reading a document he/she is presented with multiple choice *definition questions* where the task is to select the most appropriate definition for a focus-word that the student read in the document, from the given set of five definitions. The correct definition and the four distractor definitions are selected from a different ESL dictionary namely, the Longmans Dictionary of Contemporary English (LDOCE) [3]. Using two dictionaries allows us to measure the student's ability to transfer his/her learning. This however requires that the two dictionary definitions be *aligned*, that is, for each word-definition in CALD the corresponding LDOCE word-definition(s) that conveys the similar meaning have to be known. Our work in direction of dictionary definition alignment using unsupervised machine learning techniques is described in this paper.

2. Methodology

The problem of definition alignment can be transformed into that of clustering *polysemous* (same/similar meaning) definitions. As a result the requirement of learning an alignment function changes to learning similarity function. Formulating the problem in similarity space allows us to use traditional lexical similarity measures such as word-overlap, cosine similarity which are described in the Section 2.2. From a different perspective the definition alignment problem can also be viewed as a task of separating *homonym* (distinct meaning) definitions into different groups where the alignment function would get transformed to a distance function.

For example given the definitions for the word *grant* from CALD and LDOCE in Figure 1 our goal is to group these definitions such that 3 groups, each containing the polysemous definitions ({1,a},{2,b},{3,c}) from the two dictionaries are created.

CALD Definitions:
1. a sum of money given especially by the government to a person or organization for a special purpose
2. to give or allow someone something, usually in an official way
3. to accept that something is true, often before expressing an opposite opinion

LDOCE Definitions:
a. an amount of money given to someone, especially by the government for a particular purpose
b. to give someone something that they have asked for, especially official permission to do something
c. to admit that something is true although it does not make much difference to your opinion

Figure 1 *CALD and LDOCE definitions for the word "grant"*

The following sub-sections provide the details about the data and the methodology.

2.1. Data Description

The dataset consists of 383 definitions for 80 words from CALD and LDOCE. The gold standard was created by manually grouping the 383 definitions into 192 polysemy groups. Although the definition and group numbers have been specified in total the grouping was done at word-level, that is, a definition for *word1* from either of the dictionaries was never grouped with definition for *word2,* from either of the dictionaries. 90 groups have 2 data-points, 33 groups have 3 data-points, 6 groups have 4 data-points, 3 groups have 5 data-points and 1 class has 6 data-points and 59 groups have 1 data-point. The majority of the single data-point groups consist of definitions from LDOCE which indicates that LDOCE has better word-sense coverage than CALD. On an average each definition consists of 12 words.

2.2. Features

Each data-point in this task is a short sentence or phrase. Since both dictionaries, CALD and LDOCE are specifically designed for ESL students the vocabulary used by the lexicographers to author the definitions is restricted or controlled. The exact information about how much the two controlled vocabulary sets overlap or differ is not available. Nonetheless this provided a motivation for using lexical similarity measures for capturing the similarity between a given pair of definitions. The following feature types have been experimented with:

1. Raw word-overlap w/ and w/o stopwords;
2. Normalized word overlap w/ and w/o stopwords;
3. Cosine similarity w/ and w/o stopwords

The raw word-overlap captures the number of words common to both the definitions of the pair under consideration. The assumption here is that larger the number of common words between the two definitions greater is the chance of them being polysemous.

The normalized word-overlap feature bounds the overlap scores to the range of [0,1] by scaling with respect to the definitions' length (in words). Performing normalization w.r.t. the definitions' length provides a principled way of removing the bias towards longer definitions and also makes the similarity scores easily understandable and comparable. For example, it might not be obvious that an overlap of 4 words between definitions with lengths 10 and 12 is smaller than an overlap of 3 words between definitions of lengths 7 each. The following formulation was used to compute the normalized word-overlap (*nwo*) score between definitions d^a and d^b:

$$nwo(d^a, d^b) = \frac{2 \times \frac{rwo(d^a, d^b)}{|d^a|} \times \frac{rwo(d^a, d^b)}{|d^b|}}{\left(\frac{rwo(d^a, d^b)}{|d^a|} + \frac{rwo(d^a, d^b)}{|d^b|} \right)}$$

where, *rwo* stands for the raw word overlap count between definitions d^a and d^b and $|d^x|$ gives the length of the definition *x* in words. The multiplier (2) in the numerator takes care of scaling the score to 1 in the best case of complete overlap.

The cosine similarity score is computed by first representing each definition in terms of unique words that occur in the definitions (vocabulary of the dataset: *n*), this is often referred to as the "bag-of-words" approach since the order of occurrence of words is not captured in this type of representation of the data-points. The resulting definition representation can be viewed as a vector where each unique word corresponds to a dimension and each definition vector has some magnitude (0 or more) in each of these dimensions. Once these definition vectors are created the cosine similarity score for each definition pair is computed using the following formulation:

$$\cos(d^a, d^b) = \frac{\sum_{i=1}^{n} d_i^a \times d_i^b}{\sqrt{\sum_{i=1}^{n} \left(d_i^a\right)^2} \times \sqrt{\sum_{i=1}^{n} \left(d_i^b\right)^2}}$$

where, *n* is the dimension of each of the definition vectors d^x.

All the above feature types are lexically motivated, in other words, any form of syntactic information is not explicitly used. The utility of the closed-class words such as articles and prepositions (stopwords) in combination to the above lexical features was experimented by using each of the feature types with and without stopwords. Keeping in mind the short lengths of our data-points, a conservative stop-list (list of stopwords) consisting only of the following function words: articles (*a, an, the*) and prepositions (*of, to, in, for, on, with, as, by, at, from*) and an auxiliary verb *be* was used.

It is important to note that each of the above features is symmetric, that is, $feat(d^a,d^b) = feat(d^b,d^a)$. As a result each of them can be represented as a symmetric adjacency matrix *W* of *mXm* dimensions where *m* is the number of definitions to be grouped.

2.3. Algorithms

Different types of unsupervised clustering algorithms to group the definitions have been experimented. The inherent data sparsity in this task has guided most of the experimental design choices made. As specified in section 2.1 majority of the groups contain less than 5 definitions. This pattern or property of low definition density in clusters is not specific to the current dataset but is a property of the task. Thus datasets with larger number of words would not change this scenario. It is difficult to learn reliable classification models using supervised machine learning algorithms when only small amount of training data is available. This is so because such models do not generalize well on unseen data, in other words, they *overfit* the training data [5]. Taking these issues into consideration supervised machine learning algorithms were not experimented with.

The building-blocks of our experimental design are K-means (flat-clustering), Hierarchical clustering and Spectral clustering. The following combinations of these three algorithms have been explored:

1. K-means [6].
2. Spectral clustering (Ng et. al [8]) followed by K-means.
3. Spectral clustering followed by Hierarchical clustering (Ward's algorithm [7]).
4. Spectral clustering followed by Spectral clustering followed by K-means.
5. Spectral clustering followed by Spectral clustering followed by Hierarchical clustering.

K-means algorithm [6] starts with *k* random cluster means, where *k* is specified by the user and all the data-points (here definitions) are assigned to the closest cluster mean. The

Table 1 Results in terms of clustering error

Algorithm	raw word-overlap with stopwords	raw word-overlap w/o stopwords	normalized word-overlap with stopwords	normalized word-overlap w/o stopwords	cosine with stopwords	cosine w/o stopwords
K-means	0.290	0.245	0.305	**0.243**	0.290	0.258
NJW-Kmeans	0.316	0.331	0.232	0.201	0.245	**0.193**
NJW-Ward	0.311	0.324	0.230	0.198	0.240	**0.188**
NJW-NJW-Kmeans	0.284	0.298	0.251	0.214	0.242	**0.211**
NJW-NJW-Ward	0.290	0.308	0.253	0.217	0.248	**0.211**

definition of closeness used here is cosine similarity. Next, each of the data-point is re-assigned to a cluster if doing so improves the overall similarity score. The cluster mean is recomputed every time a new data-point is assigned to that cluster. The re-assigning process is repeated until no more re-assignments occur or 100 times, to avoid local optimums

The Ward's algorithm [7] starts with each data-point (definition) in its own cluster and at every step merges a pair of clusters that leads to minimal loss in information, which is measured by the error sum-of-squares criterion. This process yields a taxonomy of clusters.

The spectral clustering algorithm proposed by Ng et. al [8] first transforms the higher dimensional feature vectors (m) to a lower spectral dimension (k) and then clusters the lower dimensional data. More specifically, given a symmetric similarity/affinity matrix (W) of mXm dimensions, a diagonal matrix (D), which is sum of every row of the affinity matrix placed along the diagonal, is computed. A Laplacian matrix ($L = D^{-1/2}W\ D^{-1/2}$) is computed and its eigen-components are computed. The eigenvectors corresponding to the top k eigenvalues are selected to be represented as columns of a new matrix (X) and then the rows of X are normalized to have unit length. The rows of this normalized matrix are now clustered as one would cluster the original data-points; however, the dimension of each of the new vector is k and not m. The top eigenvectors correspond to the dimensions of largest variance, that is, the dimensions along which the most information is present. The lower-ranked vectors are typically viewed as containing noise or insignificant amount of information and thus discarding these dimensions leads to purer clusters.

Hence forth the above method will be referred to as *NJW*. As describe above NJW consists of two steps: dimensionality reduction and then clustering in the reduced dimensions. We experiment with K-means and Ward clustering in the second step of the NJW. We have also re-applied *NJW* in the spectral dimension with the intention of investigating if applying *NJW* in the reduced spectral space further adds any value. This experimental setup is based on [6].

Each of the above algorithms is applied to the adjacency matrix created by each of the above described feature set separately to compare the effectiveness of each of the feature set and the algorithm. The number of clusters were set manually in all these experiments. Automating the choice of number of clusters will be a part of the future work.

3. Results and Discussion

Table 1 presents the results of the five clustering algorithms (rows) when used with the six feature types (columns). The results are in terms of the clustering error which is a ratio of number of misclassified definitions and the total number of definitions (383). For the feature type of raw word-overlap the spectral clustering algorithm *NJW* does not improve the performance, in fact directly using the K-means algorithm is most effective. As described earlier Spectral clustering uses Eigen system and thus normalization of the data has a positive effect on the clustering quality.

When using the normalized word-overlap feature it is evident that transforming to lower dimensions using spectral clustering (*NJW*) before applying either, flat-clustering (K-means) or hierarchical clustering (Ward) is useful.

K-means is known to find locally optimal clustering solution while hierarchical clustering strives for globally optimal clustering solution the effect of which is reflected in their performances. Performing multiple iterations of spectral clustering over-applies the dimensionality reduction methodology and leads to loss of information and not just noise and thus overall does not help.

The efficacy of the cosine similarity feature type indicates that it is useful to represent the definitions in terms of the datasets vocabulary and measure the similarity along these dimensions instead of simply pooling together the word-overlap counts along any of the dimensions (among any words). K-means assumes that the data to be clustered is normally distributed at each cluster. The large reduction in the clustering error when using *NJW*, K-means and cosine instead of simply K-means and cosine indicates that the transformation caused by *NJW* makes the data Normal or near-Normal in the lower dimensions.

Overall it is evident that applying spectral clustering and thus re-representing the definitions in the reduced space in terms of their eigenvectors does help. Pre-processing the definitions to remove the stopwords all shows positive effect.

Evaluating clustering performance is often tricky because clustering is a very subjective task and thus any gold standard is unlikely to be universally accepted. Also since the gold standard was created by a single coder, inter-coder agreement is not available. Thus to gain further insight about our above results we examine the best case *NWC* (*NJW*-Ward + cosine without stopwords) in more details.

The metric used for computing the clustering error reported in the above tables penalizes cases where members of originally one cluster were split by the clustering algorithm into 2 or more pure clusters. Instead, if we measure the performance of the clustering algorithm in terms of the impurity of the generated clusters then for the best case *NWC*, the error drops down to 0.1201 (46/383), where 46 is the number of definitions that made an otherwise pure cluster impure, i.e., all the definitions that do not belong to the majority group within their cluster. The

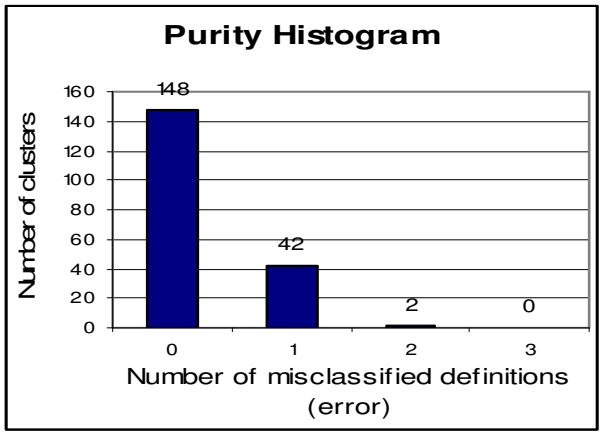

Figure 2 Purity Histogram for *NWC*

histogram of purity of clusters is shown in Figure 2. This figure shows that 148 clusters were totally pure, i.e., had zero misclassified definition, 42 clusters had one misclassified definition, 2 clusters had two misclassified definitions and none of the clusters had more than two misclassified definition.

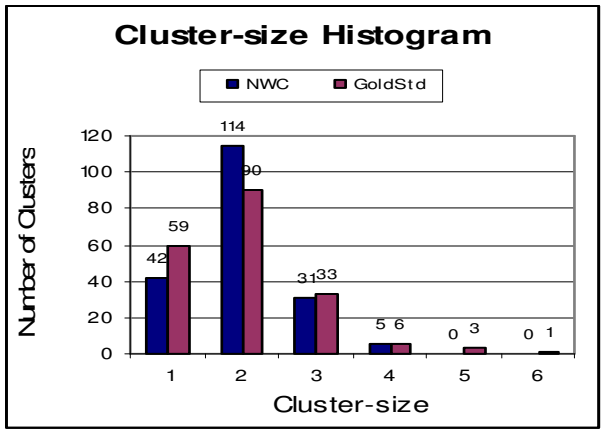

Figure 3 Cluster-size Histogram

To get a complete picture Figure 3 provides the histogram of cluster-size of both, the clustering solution given by the *NWC* and the gold standard. The plot shows that the proposed clustering solution by the best case comes quite close to the gold standard cluster-size wise too. However *NWC* seems to be struggling with cluster sizes greater than four. Although, for the task of clustering definitions the size of a cluster would typical not exceed five or six definitions, finding an elegant solution for this problem will be a part of the future work. We can also see that *NWC* has confused few (17 definitions) of the single element clusters by combining them into larger clusters. This is another direction of the future work – to find feature types which will be able to capture better discriminating features to avoid such groupings.

We also plan to look at options which might help us enrich or expand our terse definitions and thus help us build richer definition representation.

In a related work [4] the LDOCE and WordNet definitions are merged based on lexical overlap among two definitions. They exploit the taxonomy structure of WordNet for the words for which no WordNet definition is available by including definitions for synonyms and/or parent nodes. An accuracy of 90% is achieved on words with exactly two definitions in both the resources and an accuracy of 80% is achieved for words with five or more definitions.

4. Conclusions

This work shows that lexical features when filtered with an appropriate stop-list can be effectively used to represent dictionary definitions. We also show that the layered approach of spectral clustering, when followed by hierarchical clustering works better than traditional single phase clustering methods for this task. On the whole we show that the task of grouping polysemous definitions can be effectively automated using the approach described here.

5. Acknowledgements

This work has been supported by the Barbara Lazarus Women@IT Fellowship, NSF grant IIS-0096139 and Dept. of Education grant R305G03123. Any opinions, findings, conclusions or recommendations expressed in this material are the authors', and do not necessarily reflect those of the sponsors.

6. References

[1] Heilman M., Collins-Thompson K., Callan J. & Eskenazi M., "Classroom success of an Intelligent Tutoring System for lexical practice and reading comprehension." Proceedings of the Ninth International Conference on Spoken Language Processing. 2006.

[2] Cambridge Advanced Learners Dictionary: http://dictionary.cambridge.org/

[3] Longmans Dictionary of Contemporary English: http://www.ldoceonline.com/

[4] Knight, K. and Luk, L., "Building a large-scale knowledge base for Machine Translation." Proceedings of 12-th conference on Artificial Intelligence, 773-778, 1994.

[5] Mitchell T. Machine Learning. The McGraw-Hill Companies, Inc. pp. 231-236, 1997.

[6] MacQueen J. B., "Some Methods for classification and Analysis of Multivariate Observations", Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1967.

[7] Ward J. H. "Hierarchical grouping to optimize an objective function." Journal of American Statistical Association, 58(301), 236-244, 1963

[8] Ng A., Jordan M., and Weiss Y. "On spectral clustering: Analysis and an algorithm." Advances in Neural Information Processing Systems, 2001.

[9] Verma D. and Meila M. "A comparison of spectral clustering algorithms." Technical Report 03-05-01, Department of Computer Science and Engineering, University of Washington, 2003.