Clustering Word Pairs to Answer Analogy Questions

Ergun Biçici and Deniz Yuret

Koç University
Rumelifeneri Yolu 34450
Sariyer Istanbul, Turkey

Abstract. We focus on answering word analogy questions by using clustering techniques. The increased performance in answering word similarity questions can have many possible applications, including question answering and information retrieval. We present an analysis of clustering algorithms' performance on answering word similarity questions. This paper's contributions can be summarized as: (i) casting the problem of solving word analogy questions as an instance of learning clusterings of data and measuring the effectiveness of prominent clustering techniques in learning semantic relations; (ii) devising a heuristic approach to combine the results of different clusterings for the purpose of distinctly separating word pair semantics; (iii) answering SAT-type word similarity questions using our technique.

1 Introduction

This paper focuses on answering word analogy questions by using clustering. Discovering the semantic relations between pairs of words is an important problem in natural language processing, information retrieval, and question answering. We demonstrate that word pair analogies such as *hand:palm::foot:sole* given in Aristotelian format contain shared semantic relations which can be learned.

Analogy identification is an important problem when answering questions. We are often interested in the same semantic relations that hold between word pairs. Many cognitive tasks are based on analogies and analogical reasoning. Human cognition processes known as analogy derivation, analogical reasoning, and similarity judgments take central role in reasoning. The ability to answer word analogy questions is directly related to word sense disambiguation, information extraction and question answering, information retrieval, and machine learning.

Identification of the relation between words is not a trivial problem. Most of the time, we even do not have the vocabulary for representing the relations. When we are given a pair of words and asked the question of what the relation between them is, the first reaction we would give is to use a dictionary. However, even with the presence of a dictionary and knowing which sense they are used with, we may not know the semantic relation between them as it is usually context dependent. There are also open problems in identifying what a semantic relation is, how we can represent the meaning of words, and what can be the possible set of attributes for representing semantic relations. On top of this, given a set of attributes for representing semantic relations how can we learn or identify semantic relations and which method of learning to choose is also a problem.

In a vector space model, we observe arbitrarily shaped clusters of semantically related word pairs and use these clusters to obtain information beyond the distance between word pair vectors. The contributions of this paper are three-fold: (i) We cast the problem of solving word analogy questions as an instance of learning clusterings of data and measure the effectiveness of prominent clustering techniques in learning semantic relations. (ii) We devise a heuristic approach to combine the results of different clusterings for the purpose of distinctly separating word pair semantics. (iii) We answer SAT-type word similarity questions using our technique.

This paper is organized as follows. Next, we briefly talk about related work. Section 2 introduces our approach, the steps involved, our clustering techniques, and our scoring algorithm. In Section 3, we present the results of our experiments with college-level multiple-choice word analogy questions and the evaluation of our results. We discuss and present future work and conclude in Sections 4 and 5.

Related Work:

Previous approaches used vector space models [SWY75] and latent semantic analysis [LD97] to represent meaning of words. Vector space model measures the similarity of word pairs by the cosine of the angle between the feature vectors of each pair. Latent Semantic Analysis (LSA) is an application of the vector space model where feature vectors are found by the log and entropy transformations [LD97].

In a recent work by [Tur05], Latent Relational Analysis (LRA) is introduced, which extends the vector space model making use of automatic pattern generation from a given corpus, singular value decomposition (SVD), and the information of sense similarity.

2 Approach

Learning semantic relations between words is problematic since we do not have the lexicon or attributes for representing the relations. Therefore, even defining the semantic relations can be ambiguous. This paper makes the assumption that semantic relations are determined by both the word pairs and the syntactic patterns that they are observed with. The meaning of the words in the word pair that participate in the semantic relation also plays a central role in this process.

Given a word pair, $w_1 : w_2$, we may observe a semantic relation within or not and the semantic relation observed within can vary based on the syntactic patterns the word pair is observed in as well. For instance, the semantic relation in the word pair, $gold : ship$, may be different in $[ship$ `made of` $gold]$ and in $[ship$ `carrying` $gold]$.

The following examples from Pustejovski [Pus91] gives an illustration: in the sentence "the woman put out her cigarette", the semantic relation in the word pair `put out:cigarette` is *finishing-the-act-of-smoking* or at a more general level of understanding, *finishing-an-activity*. In the sentence "the man finished his beer", the semantic relation in the word pair `finish:beer` is *finishing-the-act-of-drinking* or at a more general level of understanding, *finishing-an-activity*. Figure 1 shows many levels of the senses that can be understood. The lower levels of the semantic relations have smaller number of instances or objects in the given context and can be seen less frequently in a given corpus of text. One premise of this work is that the *many levels of semantic*

relations can be discovered by applying machine learning algorithms such as clustering for different levels of confidence.

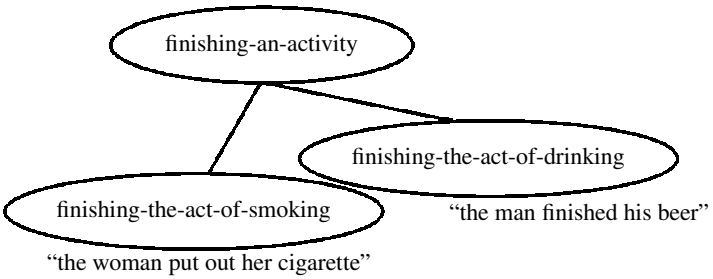

Fig. 1. Many levels of semantic relations.

Also, word pairs and their clusters obtained by finding other pairs of similar words to the original word pair can help in this analysis. For instance, the cluster obtained from the word pair `bank:city` may contain other word pairs such as `bank:town` and `bank:country` and by using many word pairs observed in clusters, we can find higher level semantic relations. Therefore, better semantic relation classification mechanisms and better word similarity judgments can be mutually useful.

In the following section, we describe the steps involved in our approach in which we use clustering techniques for finding groups of word pairs that participate in similar relations. The resulting clusters are then used for answering word similarity questions.

2.1 Steps Involved

We represent each word pair by the syntactic patterns that are observable between the words in the pair given a large corpus. For our experiments, we used the dataset from [Tur05] derived by using Waterloo MultiText System [CCB95] as the search engine.

The dataset is further smoothed mapping its feature vectors into a lower dimensional space using SVD (choosing the largest s singular values). Alternate pairs are generated by using synonymous words based on a thesaurus. However, as we have mentioned in the introduction, this may not lead to better measurements as the semantic relation that the pair discloses might change. We perform experiments regarding this choice in the next section.

Our approach is a multi-step process starting with the preprocessing steps given in [Tur05]. The input to the system is a set of word pairs, WP, that we are interested in. Each word pair, $wp \in WP$, is represented as $w_1 : w_2$ where w_1 and w_2 are the two words in the pair. We give an overview of the steps involved below:

1. **Identify alternates:** For all $wp \in WP$, find alternate pairs **wp**. An alternate pair is generated by changing each word in the pair with one of its top 10 synonyms that can be found in the thesaurus. Thus,

$$\mathbf{wp} = \bigcup w_1' : w_2',$$

where w'_1 is any one of the top 10 synonyms of the word w_1.

2. **Select alternates:** For each $wp' \in$ **wp**, query the search engine for patterns of the form $[w'_1 * * * w'_2]$, where $wp' = w'_1 : w'_2$. Sort **wp** based on the frequency of the formed phrases and select the top 3 most frequent alternate word pairs. Let **wp**$'$ be the set formed by the union of the top 3 most frequent alternate pairs and the original word pair, wp.
3. **Find patterns:** For all $wp \in$ **WP**, where **WP** $= \bigcup$ **wp**$'$, query the search engine to find patterns of the form $[w_1 * * * w_2]$, $[w_1 * *w_2]$, or $[w_1 * w_2]$. Each $*$ in the pattern can match a word from the corpus. Select the top 4000 patterns.
4. **Generate a matrix:** For each $wp \in$ **WP**, create a row and for each pattern $w_1\ P\ w_2$, create a column for $w_1\ P\ w_2$ and another one for $w_2\ P\ w_1$. Thus, we will have 8000 columns. In the final matrix, X, $X(i, j) =$ *frequency of the* j*th pattern that contain* i*th word pair.* Feature vectors are found by the log and entropy transformations [LD97]. X is a sparse matrix.
5. **Apply SVD:** The matrix, X, is further smoothed by mapping its feature set to a lower dimensional space using SVD [TB97] by choosing the largest 300 singular values to reduce the number of columns to 300 instead of 8000. Let this new matrix be X_{300}.
6. **Apply clustering:** We apply k-means, and spectral clustering on X_{300}. This provides us with different clusterings (i.e. allocation of word pairs into disjoint clusters).
7. **Apply scoring function:** The resulting clusterings are scored and combined to answer analogy questions and to pick the correct answer from a given set of choices.

In the following section, we introduce the clustering techniques we used and our scoring function.

2.2 Clustering for Semantic Relations Between Word Pairs

We used two different clustering techniques for analysing our data: (1) k-means clustering, which partitions the data of N points into k clusters where each cluster is represented by the center of gravity of the cluster; (2) spectral clustering [AYN01], which clusters points using eigenvectors of matrices derived from the data (i.e. using the relative distances of points in the similarity matrix of the data).

Local scaling is a technique which makes use of the local statistics of the data to separate the clusters. This is done by scaling each point in the dataset with a factor proportional to its distance to its kth neighbor. We used a locally scaling version of spectral clustering [ZMP04] for our experiments.

The usual definition of clustering assumes that clusters are regions in the space such that the intra-similarities of objects in them are maximized and the inter-similarities between them is minimized. However, in high dimensional spaces the measures of closeness based on the Euclidean distance between objects are not always appropriate.

2.3 Scoring Algorithm:

Both k-means and spectral clustering take k, the number of clusters, as input. We experimented with k values $2^1, 2^2, \ldots, 2^8$. How to score according to different k clusterings

is an issue as each one represents a different view of the data. This section presents our approach to combining different clustering results for the task of answering word pair analogy questions with each one having 5 choices. Each *clustering* is an allocation of points to different clusters based on k.

Input: qwp: A question word pair, AWP: A set of answer word pairs, $clusters(wp)$: a function which returns the clusters for a given word pair, $numClusterings$: number of clusterings, $numofClusters[cluster]$: number of clusters for a given clustering $cluster$.
Output: An answer choice which attains the highest score.

$cq = clusters(qwp)$;
for $cluster = 0;\ cluster < numClusterings;\ cluster++$ **do**
    $ns = 0$; /* Number of distinct answers in the same cluster as the question */
    foreach $awp \in AWP$ **do**
        $ca = clusters(awp)$;
        if $cq[cluster] == ca[cluster]$ **then**
            $ns++$;
        end
    end
    foreach $awp \in AWP$ **do**
        $ca = clusters(awp)$;
        if $cq[cluster] == ca[cluster]$ **then**
            $score[awp] = score[awp] + numofClusters[cluster]/ns$;
        end
    end
end
$choice = max(score)$;

Algorithm 1: Clustering Scoring Algorithm.

Each score is weighted according to the number of clusters, n_{c_i}, the clustering c_i has formed. The number n_{c_i} is a measure of the precision of each cluster formed. We call the number of cases a clustering disjointly clusters the question and an answer separately from other answers, $nDisjoint$. This number is also used for reaching the final score of each alternative.

There are many decision points for the scoring function: (i) What to do in case of ties in the score? (ii) Can we benefit from alternate word pairs? Alternations in questions? Alternations in answers? Both? (iii) Should we take the average score or the maximum?

With our scoring method, given in Algorithm 1, each clustering learned can contribute to the final classification independent of others. Thus, we may even combine clusterings resulting from different clustering techniques.

3 Experiments with SAT Questions

We used 374 college-level multiple-choice SAT analogy questions to test our approach, which makes the total number of word pairs used 8128. This number is found after

adding alternate pairs for each original question word pair ($(374 \times 6 \times 4)-$ no alternate cases $-$ word pairs not present in any pattern). In our experiments, we used the dataset generated by using the Waterloo MultiText System. For generating alternate pairs, Lin's [Lin98] automatically generated thesaurus was used. For SVD, we used Rohde's SVDLIBC, which is based on [Ber92].

SAT-questions dataset contais rare word pairs, in which the average human score is 57% [Tur05]. Hence, the dataset might not be effective in discriminating the semantics of word pairs. Another problem might be our distance measure. The cosine distance between word pair vectors may not be a good measure of distance for clustering the semantic relations. We may need to use other distance measures like the Euclidean distance for clustering.

We are using weighting to combine the results of a clustering to get a single score. We have experimented with different weighting metrics for each clustering result. One choice is to give each clustering equal weight and another is to weigh them by the number of clusters in each. We are using the latter which gave better results.

Figure 2 illustrates the effect of using different number of clusterings on accuracy. $x = 64$ corresponds to the case where we included clusterings with $k = 2, 4, 8, \ldots, 64$. The lower curve indicates the accuracy when the program selects the answer randomly in case of a tie. The upper curve shows the accuracy when the program selects the correct answer in case of a tie.

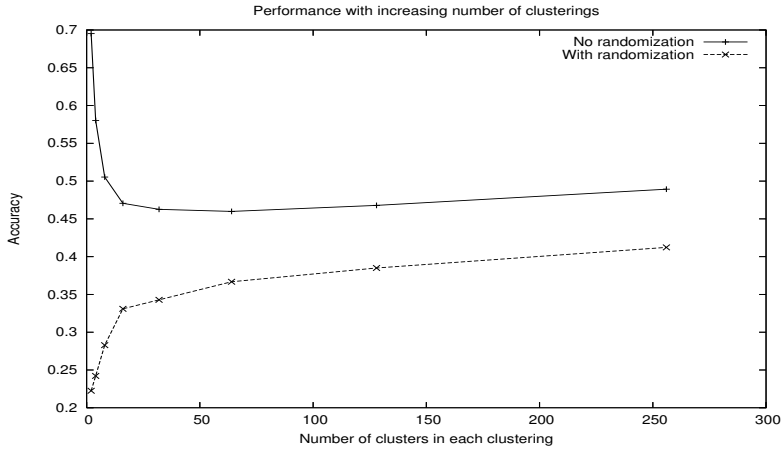

Fig. 2. Performance graph with increasing number of clusterings.

Performance of clustering methods of both k-means and spectral clustering for clustering with 256 clusters is still being computed as of this writing. in answering word analogy questions is shown in Table 1. We have tried using the thesaurus based alternates for the question pair, answer pairs, both, or none. Using alternates does not seem to consistently improve the accuracy, however note that the alternates are always used in performing the SVD and clustering.

Alternates	k-means	Spectral with local scaling
none	41.23%	35.72%
question	44.01%	34.87%
answer	39.95%	35.45%
both	38.50%	32.89%

Table 1. Performance of clustering methods in answering word analogy questions. The first column shows whether thesaurus based alternate word pairs have been used for the question and answer pairs.

4 Discussion and Future Work

Average human score on this dataset is reported to be 57% [TL05]. Turney [Tur05] reports a performance of 56% using latent relational analysis. Although clustering results in a lower performance than LRA, there are some benefits of clustering: (1) We can learn discrete semantic relations. (2) We can learn a hierarchy of semantic relations. We plan to learn the hierarchy of relational concepts that are observed in the dataset and see whether the resulting clusters are meaningful. Analysis of the clusters in terms of their conformance to known semantic relations is also part of our future work. Various ontologies [Fel98] and Nastase's set of semantic relations [NS03] form a good basis for comparison.

These SAT questions are not easy to solve on the average and the semantic relations in word pairs might not be clearly cut. Automating the task of answering similarity questions for typical word pairs may be easier in general. Future work includes automating this task for the top 5000 word pairs in a given corpus.

5 Conclusion

We present an analysis of clustering algorithms' performance on answering word similarity questions. The dataset we have is based on word pairs and their occurrence frequencies in some common sytactic patterns. We observe that semantic relations between word pairs may be distinguished by using clustering techniques.

Clustering semantically close word pairs may be more informative than a distance between word pair vectors. This paper contributes by: (i) casting the problem of solving word analogy questions as an instance of learning clusterings of data and measuring the effectiveness of prominent clustering techniques in learning semantic relations; (ii) devising a heuristic approach to combine the results of different clusterings for the purpose of distinctly separating word pair semantics; (iii) answering SAT-type word similarity questions using our technique.

Future work includes learning the hierarchy of relational concepts and observing the resulting clusters' performance with respect to already identified set of semantic relations.

Acknowledgements

We acknowledge the generous allowance of Peter Turney from NRCC, Canada, for access to the dataset and Waterloo MultiText System.

References

[AYN01] Yair Weiss Andrew Y. Ng, Michael I. Jordan. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14, 2001.

[Ber92] Michael W. Berry. Large scale singular value computations. *International Journal of Supercomputer Applications*, 6(1):13–49, 1992.

[CCB95] Charles L. A. Clarke, G. V. Cormack, and F. J. Burkowski. An algebra for structured text search and a framework for its implementation. *The Computer Journal*, 38(1):43–56, 1995.

[Fel98] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[LD97] Thomas K. Landauer and Susan T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240, 1997.

[Lin98] Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, pages 768–774, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

[NS03] Vivi Nastase and Stan Szpakowicz. Exploring noun-modifier semantic relations. In *Fifth International Workshop on Computational Semantics (IWCS-5)*, pages 285–301, Tilburg, The Netherlands, 2003.

[Pus91] James Pustejovsky. The generative lexicon. *Computational Linguistics*, 17(4):409–441, 1991.

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[TB97] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 1997.

[TL05] Peter Turney and Michael L. Littman. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1-3):251–278, 2005.

[Tur05] Peter Turney. Measuring semantic similarity by latent relational analysis. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 1136–1141, Aug 2005.

[ZMP04] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Eighteenth Annual Conference on Neural Information Processing Systems*, 2004.