Concept Based Clustering Approach for Efficiency Evaluation for Homonym Words Identification

Sunil Kumar and Rajendra Gupta
Rabindranath Tagore University, Bhopal, India

Keywords: Homonym Words, Concept Based Similarity Measure, Clustering Technique, Similarity Measures, Heteronomy.

Abstract: The word homonym is derived from the Greek word 'homonyms,' which means 'to have the same name'. The prefix 'homo' and the suffix 'nym' both indicate the same thing. Two or more words having the same spelling or sound but different meanings are called homonyms. The concepts could be words/phrases or expression and they are completely dependent on the sentence semantic organization. In this paper, there are four similarity matrices are computed utilizing the similarities found using the word-based and combination method concept analysis. For assessing the influence of concept-based similarity on clustering, three typical document clustering approaches has been analysed. Experiments shows that the method described in this research outperforms the earlier proposed methods in identifying homonyms in intersections of tag contexts using clustering algorithm which is the best way to find them. The K-nearest algorithm performs better similarity measures for concept-based homonym words.

1 INTRODUCTION

The homonymy is a common occurrence and separating their distinct meanings is a major challenge in natural language processing. The contextualized word embeddings have made it possible to distinguish token level occurrences, and they have been used for supervised word sense disambiguation, but it is unclear whether they can also capture latent meaning distinctions without labels, especially when the meaning boundaries are ambiguous Christopher et.al., 2022).

Identifying homonyms is one of the issues that make collecting and evaluating data from the scientific literature difficult. The terminology used to explain homonymy, heterography, and related phenomena is a bit muddled and often misunderstood, so some cleaning up is in order for clarity. It has at least six similar terminologies, homo-/hetero- prefixes with -phone/-nym/-graph suffixes that define a relationship that exists between a set (usually pair) of word kinds. It is essential to understand that to make correspond to combinations of identifying / difference in three parameters, sound, writing, and meaning, according to how the terms are traditionally used: the combination of same sound, same spelling, but distinct meaning is for homonyms, and so on.

Here, it is left with six options after eliminating the uninteresting combinations of the same and different in all of these ways (Vinh et.al. 2021). The correspondence is shown in the table below.

Table 1: Categorization of homo/hetro terms.

Sound	Writing	Meaning	Term	Example in English words
Same	Same	Diff	Homonym	Bank/bank
Same	Diff	Diff	Homophone	There/their
Diff	Same	Same	Homograph	Either
Diff	Same	Diff	Heteronmy/ heterophone	Bow/bow
Diff	Diff	Same	Synonym	Stop/halt
Same	Diff	Same	Heterography	Racket/racquet

Kumar, S. and Gupta, R.
Concept Based Clustering Approach for Efficiency Evaluation for Homonym Words Identification.
DOI: 10.5220/0012607900003739

In *Proceedings of the 1st International Conference on Artificial Intelligence for Internet of Things: Accelerating Innovation in Industry and Consumer Electronics (AI4IoT 2023)*, pages 153-157
ISBN: 978-989-758-661-3

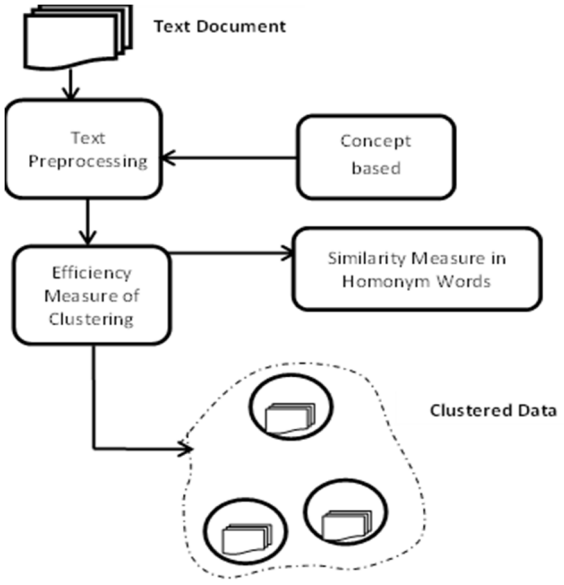

Figure 1: Concept based Similarity Measures for Homonym Words.

Homonyms and homophones are distinguished by whether or not the writing is identical (homonyms) or not (homophones) in addition to sound, but this distinction is not always necessary, thus it can be used the term homonym to refer to both unless the distinction is critical (Westgate et.al. 2021).

2 LITERATURE REVIEW

A significant number of research papers on homonym word detection have focused on a particular context, like data in a bibliographic or process model (Xu et.al., 2021). Author produced a painstakingly hand-crafted training set from the whole PubMed collection by going through numerous iterations for the same aim, and applied a three-step clustering for name disambiguation using common coauthors.

The author built a system that uses user feedback for disambiguation and provided a fix for incremental disambiguation (Xu et.al., 2020). Finally, a method is suggested for finding homonyms and other lexical difficulties in process models.

A text detection system built on an online learning community tries to automatically spot homonyms and other semantic problems. Also, the author developed a novel ambiguity measure for work (homonyms, for example) using Durda, Caron, and Buchana algorithms. The hash approach is used to quickly determine the distance in a number of algorithms for homonym mistake detection.

Therefore, a novel homonym detection method that takes into account a word's semantic meaning when the algorithm works.

When trying to find synonyms, the author in (Yu et.al., 2020) suggests TCS as one of the initial variables to consider (together with Resource Context Similarity).

Working with stems also makes it possible to combine logically related words, such as nouns, adjectives, and verbs, which are frequently used as interchangeable tags (Yu et.al., 2019). For this reason, he used a combined approach to analyses synonymies, specifically by counting how many synonyms a "universal" method like TCS can find that could have also been found using, for example, Levenshtein distance, synonym checking on Wordnet, and term translation with Wikipedia (Balazs et.al, 2019).

2.1 Clustering Based Similarity Measures for Homonym Words

The concepts are represented as knowledge units that represent a collection of perceivable items with comparable characteristics. This concept with knowledge units is unmistakable (An et.al. 2019). *The objects having various attributes are abstracted into different ideas, even though they are referred to using the same literal word. A concept system is a collection of related concepts arranged in a logical order.*

2.2 Pre-Processing

Stop words are removed during the preparation phase, tokenization is conducted, and the word vector is located in semantic space. Then find vector representations of every word in the phrase by tokenizing the text and relocating stop words (Bhardwaj et.al., 2018).

2.3 Tokenization

The tokenization is a process of breaking down phrases into tokens and removing unnecessary punctuation and other types of characters. The Vector Space Model with tf-idf weighting is used to represent documents.

The following points are based on the calculation of 'ctf' for concept called 'c' in sentence 's', where the document is denoted as 'd':

Calculating ctf of Concept c in Sentence s
The term 'ctf' indicates how frequently a concept 'c' appears in the verb argument structures of a phrase (sentence 's'). The purpose of the notion 'c', which recurs in different verb argument formulations of the same sentence 's', is to principally add to its meaning. In this case, the 'ctf' is a local metric at the phrase level.
Calculating ctf of Concept c in Document d

A concept c can have many ctf values in different sentences in the same document d. Thus, the value for ctf for concept c in the given document d is calculated as:

$$ctf = \frac{\sum_{n=1}^{sn} ctf_n}{sn}$$

where sn is the complete sentences in the given document d that include the concept c. The complete relevance of concept c to the denotation of its sentences in document d is dignified by averaging the ctf values of concept c in its sentences in document d. The total importance of each notion to the semantics of a document as indicated through sentences is calculated by averaging the ctf values.

Algorithm 1: Proposed Clustering-based Similarity Measure.

1. Notion of new document as ddoci
2. Denotation of Empty List as L (L is a concept list)
3. New sentence formation as sdoci in document ddoci
4. Building concepts list as Cdoci from New Sentence sdoci
5. for each concept ci ∈ Ci
   do
6. calculate ctfi of ci in ddoci
7. calculate tfi of ci in ddoci
8. calculate dfi of ci in ddoci
9. A seen document denoted as dk where k = {0, 1, . . . , doci–1}
10. Apply Sentence sk is in document dk
11. Building concepts list as Ck from sentence sk
12. for each concept cj □ Ck do
13. if (ci == cj) then
14. update dfi of ci
15. calculate ctfweight = average (ctfi, ctfj)
16. Addition of new concept which matches in L
17. end if
18. end for
19. end for
20. Output of the matched concepts in list L

The process of calculating the ctf, tf and df in the matched concepts from the text is designated by the concept-based measure algorithm. The procedure starts with a new document (in line 1) that has evidently specified text boundaries. Each statement gives a semantic label. For concept-based similarity calculations, the lengths of the matched concepts and its verb argument structures are saved (Buchta et.al. 2018).

The concept-based similarity measure between words with homonym words is calculated using the Conceptual Term Frequency (ctf).
Consider the following concepts,
c1 = ''w1w2w3'' and c2 = ''w1w2''

where c1, c2 are concepts and w1, w2, w3 are individual words. After removing stop words, if c2 c1, then c1 holds more conceptual information than c2. In this case, the length of c1 is used in the similarity measure between c1 and c2.

The concept length is only used to compare two concepts; it has nothing to do with determining the importance of a concept in terms of sentence semantics. The ctf is used to identify relevant ideas in terms of sentence semantics as tf.

$$sim_c(d_1, d_2) = \sum_{i=1}^{m} max\left(\frac{l_{i1}}{L_{vi1}}, \frac{l_{i2}}{L_{vi2}}\right) \times weight_{i1} \times weight_{i2}, \ldots \ldots \ldots$$

The concept-based similarity between two documents, d1 and d2 is calculated by:

$$weight_i = (tf\ weight_i + ctf\ weight_i) \times log\left(\frac{N}{df_i}\right)$$

Table 2: Clustering improvement using Concept based TF.

Data Set	Methods	Single Term	Concept based Term	Term Weight (%)
		F-measure	F-measure	
Reuter	CAC	0.705	0.844	0.20
	Single-Pass	0.475	0.640	0.45
	K-NN	0.400	0.640	0.45
MongoDB	CAC	0.635	0.702	0.35
	Single-Pass	0.600	0.722	0.34
	K-NN	0.442	0.460	0.58
20-Newsgroup	CAC	0.720	0.824	0.20
	Single-Pass	0.468	0.540	0.50
	K-NN	0.400	0.566	0.54

The F-measure of clustering is used in comparison based on Conceptual Term-Frequency. Also, the entropy is also calculated for concept-based similarity measures.

3 DATASET AND CLUSTERING METHODS IN HOMONYM IDENTIFICATION

There were three data sets in the experiment have taken to test the homonym words detection. The first batch of data includes 20,215 abstract articles gathered from MongoDB digital library. The Reuters 20,570 data collection contains 12,902 documents in the second data set. The training set has 8,653 documents and the test set taken 3,490 documents. For assessing the influence of concept-based similarity on clustering, three typical document clustering approaches were chosen:

1) Categorized Agglomerative Clustering (CAC)
2) Single-Pass Clustering
3) k-Nearest Neighbor (k-NN)

In terms of classes, the cluster with the greatest F-measure is deemed the cluster that maps to class and the F-measure becomes the class score. The weighted average of the F-measures for each class makes-up the overall F-measure for the clustering and can be denoted as :

$$F_C = \frac{\sum_i \quad (|i| \times F(i))}{\sum_i \quad |i|}$$

where $|i|$ is the number of objects in class i. Due to the higher precision of the clusters mapping to the original classes, *the higher the total F-measure, the better the clustering*.

The above Table 2 shows the comparison between Single Term, Concept based similarity for three different datasets like Reuter, MongoDB and 20-Newsgroup for the clustering algorithm CAC, Single-Pass Clustering and k-NN. The F-measure of clustering is used in comparison based on Term-Frequency. Also, the entropy is also calculated for concept-based similarity measures.

The Table 2 clearly demonstrate the K-Nearest Neighbor and CAC algorithms outperforms for the clustering of homonym words detection and identification. The results of concept-based F-measure with respect to Single Term F-measure is almost better for all three algorithm CAC, Single-Pass Clustering and k-NN. So, as per the applying concept-based similarity measure on the homonym words, the results are better.

4 CONCLUSIONS

The semantic structure of sentences in documents is used to achieve a better text clustering outcome. The first part of the analysis analyses the semantic structure of each phrase to identify the sentence concepts using the ctf metric that has been proposed. The concept-based term frequency tf is used in the second component, document-based concept analysis, to analyse each idea at the document level. The third component uses the df global metric to assess concepts at the corpus level. The result demonstrates the K-Nearest Neighbor algorithm outperforms for the clustering of homonym words detection and identification. The results of concept-based F-measure with respect to Single Term F-measure is almost better for all three algorithm CAC, Single-Pass Clustering and k-NN.

REFERENCES

Christopher J.M. and Pat Langley P.M. (2022). Efficient induction of language models via probabilistic concept formation, Proceedings of the Tenth Annual

Conference on Advances in Cognitive Systems, arXiv:2212.11937v1

Vinh N.X., Epps J. and Bailey J. (2021). Information theoretic measures for clustering comparison: variants, properties, normalization and correction for chance, J. of Mach. Learn. Res. 11:2837–2854

Westgate M.J. and Lindenmayer D.B. (2021). The difficulties of systematic reviews. Conservation Biology, 31(5):1002– 1007

Xu H., Zhang C., Hao X. and Hu Y (2021). A machine learning approach classification of deep web sources. In: Fourth I. Conf. on Fuzzy Systems and Knowledge Discovery (FSKD 2022), Vol 4. IEEE, pp 561–565

Xu H., Zhang C., Hao X. and Hu Y. (2020). A machine learning method classification of deep web sources", In:Fourth Int. Conf. on fuzzy systems and knowledge discovery (FSKD 2020), vol 4. IEEE, pp 561–565

Yu J., Li J., Yu Z. and Huang Q. (2020). Multimodal transformer with multi-view visual representation for image captioning", IEEE Trans Circuits Syst Video Technology, pp.121-123

Yu J, Tan M., Zhang H., Tao D. and Rui Y. (2019). "Categorised deep click feature prediction for fine-grained image recognition", IEEE Trans Pattern Anal Mach Intelligence, pp.1021-1023

Balazs J.A. and Velásquez J.D. (2019). Opinion mining and information fusion: a survey", Information Fusion 27:95–110

An Y., Liu S. and Wang H. (2019). Error detection in a large-scale lexical taxonomy, Information 11(2):97

Bhardwaj P. and Khosla P. (2018). Review of text mining techniques, IITM J. Manag. IT, 8(1):27–31

Buchta C., Kober M., Feinerer I., Hornik K. (2018). Spherical k-means clustering, J. of Statistical Software, 50(10):1–22