CS 657A
Information Retrieval
Document Similarity using Lexical Chains

Sahil Bansal (150614) Aviraj Mishra (150170) Mohit Duseja (150419)

April, 2018

1 Problem Introduction

Traditional document clustering algorithms consider only the syntactic relationship among words and don't take into account semantic relationships so they cannot accurately represent the meaning of documents. To overcome this problem, introducing semantic information from ontology such as WordNet has been widely used to improve the quality of text clustering. The core idea of any document clustering algorithm is to judge the similarity between the documents and cluster the documents that are similar enough. In this project, we report our attempt towards integrating WordNet with lexical chains to alleviate common challenges, such as synonymy and polysemy and that of extracting core semantics from texts.

Lexical chaining is a method to incorporate the meaning of a document in so-called lexical chains. A lexical chain contains a set of semantically related words from texts, which can represent a semantic concept of the document. We try to extract a set of such chains to cover all the concepts that the document is about. The presence and concentration of such chains can be used to judge similarities among the documents.

2 Overview of our Approach

2.1 Initializing Chains

The initial set of lexical chains are created in the following manner.Since the contexts which the document might be talking about are essentially decided only by the nouns present in the document, we initially extracted the nouns using POS tags from NLTK library. If a noun appears more than a threshold R times in the document (Say a word must appear atleast 2 times to be a part of some lexical chain), then a chain is setup for that word consisting of all its synsets. Each synset incorporates one sense of that word.

2.2 Merging Chains to create concepts

The initially created set of lexical chains are merged in two steps on the basis of following relations:

- Strong Relations : when two synsets are the same or related by hypernym/hyponym
- Weak Relations : when two synsets have between them a path similarity greater than some value α.

The process of chain merging involves two stages. First all inter-chain strong relations are resolved, followed by the weak relations. The process of resolving relations is as follows : Each existent chain is run against all other chains to determine how many strong relations exist between its synsets and those of the other chain. The first pair of lexical chains that have number of strong relations greater than some threshold is merged into a single chain and appended to the list of lexical chains while the individual two chains are dropped. This process is repeated till no more merging of chains owing to strong relations is possible.

Once all possible strong merging is done, we look for the weak relations among the chains. If two chains have been them greater than some threshold number of weak relations then they are merged in the similar manner as before. The threshold for number of weak relations is kept higher than the threshold for number of strong relations as having a strong relation represents a great sense of closeness of the synsets. Again this merging procedure is continued till no more merging due to existence of weak relations is possible. Therefore, at last we are left with some reduced

number of chains in comparison to what we had at the beginning for a document.Each chain now represents a concept or a topic that the document is talking about. This reduced set of chains act as a representative for a particular document and we refer to this set of chains as the **linked document** for a particular document.

Along with the lexical chains, we also maintain a dictionary for the nouns in a document to capture importance of each noun using its term frequency. This dictionary acts as the **document proper** for any particular document. We also calculate the document frequency of each term so that we can calculate the idf values for the terms which acts as a method to incorporate global importance of terms in the corpus.

2.3 The structure of Chain

We define a class Chains() which contains the following components:

- words : a list of the words that have been merged together in the chain.
- senses : list of the semantic senses from Wordnet that the chain represents.
- count : stores sum of counts of all words in word list
- dfreq : stores document frequency of the chain

2.4 Weighting of Document Proper and Linked Document

The document proper and linked document are treated as vectors weighted by the following formula which incorporates both the term frequency and document frequency :

$$w = \frac{tf \cdot log(df)}{\sqrt{\sum_i (tf_i \cdot log(df_i))^2}} \tag{1}$$

2.5 Computing Similarity Score

The similarity between two documents is computed using two factors:

- First the document propers which represent dictionary of a document are iterated over and if there is a common word between the two dictionaries, then we update the similarity score by adding the product of normalized count of the word in two dictionaries. At last what we get represents the similarity score contributed by the vector space model.
- The second similarity score is computed using the linked document for two documents which represents the list of chains in the respective documents. To compute the similarity score, we iterate over all the chains in two documents. Suppose document A has m chains and document b has n chains. We take i^{th} chain of document A and j^{th} chain of document B. Now, suppose i^{th} chain of A has k_i synsets and j^{th} of B has k_j synsets, then we iterate over all of them one by one and if there is a common synset then we update the similarity score by adding the product of chain counts for two chains (in this case counts of i^{th} chain of A and j^{th} chain of B). At last what we get represents the similarity score contributed by the lexical chains (let's call it the semantic similarity score).

Since there will be words in documents that are absent from Wordnet, so to capture their effects towards similarity score our model needs to take part of the vector space model as well. Hence

$$\text{Sim(A,B)} = \alpha \cdot \text{vector space similarity} + (1 - \alpha) \cdot \text{semantic similarity} \tag{2}$$

The parameter α is tuned to obtain more accurate model.

3 Experiment and Evaluation

3.1 Dataset used : Wiki10+ Dataset

Wiki10+ is a dataset created during April 2009 with data retrieved from the social bookmarking site Delicious and Wikipedia. It is available for research purposes.It contains english wikipedia articles with at least 10 annotations on Delicious. Therefore, the tag information for each of these Wikipedia articles as well as the text content can be found in this dataset.

3.2 Testing our model

We took a set of documents from this corpus and tested our model to check for document similarity among any two documents from our corpus. After some efforts in parameter tuning, following parameters were found to give good results for this dataset:

- threshold number of strong relations = 1
- path similarity for strong relation = 1
- threshold number of weak relations = 2
- minimum path similarity of weak relation = 0.2

As intended, our model gives better similarity scores for the pair of documents having common tags as compared to the vector space model (for this, we manually extracted the text of documents from our corpus having common tags and tried our model on it). This can be explained by the following argument: since the tags are assigned keeping in mind the semantics as well, the tags in their own way also try to capture the topics the document is talking about which in our case is done by lexical chains. This is not taken care of in traditional vector space model, thereby giving poorer results.

The document text that clearly show distinction between the two models i.e. vector space model and our proposed model have been added to the submission. These particular examples clearly show that our model is far better in identifying and judging similarity based on semantics as can be seen in the particular case of polysemy : moviestar and skystar in our submission, where vector space model assigns similarity score of 0.596700477691 , while our model assigns it a similarity score of 0.141501988162. Also in other cases, for eg. the documents 'Belgian' and 'GreatDane' talk about these respective breeds of dogs. Since the names of breed are different, the vector space model assigns comparatively lesser similarity score (0.783811141249) as compared to the score assigned by our model (0.820084197582).

4 Qualitative Evaluation of our Document Similarity Model

Our main motive for using lexical chains rather than the simpler vector space model is its possibility of producing semantically richer results. Our results are evidence of the same. We also constructed vector space model side by side to compare its results with our model. We saw examples of lexical chaining incorporating both polysemy and synonymy.

4.1 Polysemy

To test for polysemy, we introduced two documents to our corpus, a Wikipedia article on movie-stars and another on astronomical stars. The word "star" naturally occurred heavily in both of the documents. However, "star" is used in its astronomical sense in the first article and in its popular media sense in the other. The vector-space model, unable to disambiguate the sense of the word "star", shows high similarity between the two documents. Our lexical chaining model on the other hand is able to distinguish between the different senses in which the word is being used and thus shows very low similarity between the two.
In fact, if we examine the resulting synsets, we see that in joining with words like "Sun" in the astro-stars article, the word "star" retains only the synset *Noun-7754660: "(astronomy) a celestial body of hot gases"*; for movie-stars, it has become part of a chain including the words "actor," "player," and (Bruce) "Lee" and acquired the synset *Noun-8024371: "a theatrical performer."* Thus while calculating the similarity score for the two documents, these synsets do not match and hence the appearence of the word "star" does not add up to the similarity score.

4.2 Synonymy and semantic similarity

A naive vector space system suffers also from a failure to identify words that are distinct but semantically very similar. With lexically-chained documents, performing a linked comparison, which consists of looking for synsets in the documents that are close in the Wordnet graph allows us to capture those relations that are more subtle.

5 Conclusions and Future Work

We have compared the performance of a baseline vector-space model with a lexical chaining-based algorithm for determining document similarity. The latter approach implements a hybrid scheme, where recognized words are treated with semantic awareness, while unrecognized terms are handled using a standard vector-space-like tf-idf model. Our results clearly indicate the potential of lexical chaining for incorporating some level of semantic awareness, e.g. polysemy and synonymy, into document similarity measures.

One area for improvement is to tune the lexical chaining implementation. The model involves a few parameters, tuning all of which together is quite challenging. Implementing better methods to tune the parameters could further increase the accuracy of our model.

Another area of possible exploration relating to this is to apply lexical chaining to support semantically-aware queries i.e. determining to which paragraphs in the collection the queries are most closely related, as well as treating the query as a separate document from which chains and synsets are extracted before similarity determination.

Note: Instructions for running the system are included in the submission's README file.

References

[1] Dataset. http://nlp.uned.es/social-tagging/wiki10+/.

[2] S. S. Desai and J. Laxminarayana. Wordnet and semantic similarity based approach for document clustering. In *Computation System and Information Technology for Sustainable Solutions (CSITSS), International Conference on*, pages 312–317. IEEE, 2016.

[3] D. Jayarajan, D. Deodhare, and B. Ravindran. Lexical chains as document features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008.

[4] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.

[5] T. Wei, Y. Lu, H. Chang, Q. Zhou, and X. Bao. A semantic approach for text clustering using wordnet and lexical chains. *Expert Systems with Applications*, 42(4):2264–2275, 2015.