Representation of Text Document using Wordnet and Lexical Chains

Geethalekshmy V, Reshma R, Anjali B
Department of Computer Science and Application,
Amrita Viswa Vidyapeetham,
Amritapuri, India
geethalekshmy@am.amrita.edu, reshmaram008@gmail.com, anjaliusha29@gmail.com

***Abstract*— This paper aims at comparing the documents with the help of Lexical chains to check how similar/dissimilar the documents are. Lexical chains help to identify the main theme of the document. Lexical chains corresponding to a document capture the main structure and content of the document. Initially, word sense disambiguation(WSD) is done on a document to identify the sense of words that is present in the document. WSD is done using a lexical database named Wordnet. The nouns of a document are replaced by appropriate word senses according to a particular context. These word senses are used for constructing lexical chains. Word senses which are related to each other with hypernym or meronym relation are added to a lexical chain. There may be multiple lexical chains for representing parts of a document and the chain with a high score will give an idea about the main theme of the document. Comparing these highest scored lexical chains of each document will help to find out the similarity between them.**

I. INTRODUCTION

Lexical chains help to represent the basic theme of a document. Lexical chains are built on the basis of disambiguated semantic concepts. Each lexical chain may have its own score and the score is found out by adding the weight of each concept in that particular lexical chain. To find the similarity between the documents, the lexical chain of each document having highest score is compared. For constructing lexical chains, WSD must be initially performed. It extracts all nouns from a document and replaces it with the most appropriate sense. These word senses are used for constructing lexical chains. For the construction of lexical chains, there exist semantic relations called hypernym and meronym. If the word senses are related to each other with hypernym or meronym relation, lexical chains are constructed. Each document can have multiple lexical chains for representing different parts of a document. The lexical chain having the highest score gives an idea about the basic theme of a document. We cannot build lexical chains directly from a document. The two steps involved in document representation are:

1. Performing Word Sense Disambiguation to identify which sense/concepts of a word is used in a document.
2. Constructing lexical chains using the concepts.

Lexical chains are identified by using the relationship between the senses of the words. Each node in a lexical chain is a word sense of a word, and each link can be hypernym or meronym relation between two-word senses. Since lexical chains are built on the basis of disambiguated semantic concepts, it adjusts weights of concepts in each lexical chain by adding a weight based on relations that a particular concept has with other concepts. At last the weights of concepts in a lexical chain are added together to arrive at the score of this lexical chain and when the score of a lexical chain passes the pre-defined requirement, the concepts in it are added to the core semantic feature set. The lexical chain with highest score represents the theme of a document. To find out the similarity between two documents, the lexical chains with highest score of each document is taken and the nodes of the lexical chains are compared to check how similar/ dissimilar they are. To perform both

these operations WordNet has been used. It is one of the largest, widely used lexical English database. It normally resembles a thesaurus, in that it groups words together based on their meanings. The grouping of words doesnt follow any explicit pattern other than the meaning similarity.

A. Performing Word Sense Disambiguation

Word sense disambiguation uses most appropriate sense by replacing the original terms in a document. Here, it is done using the predefined functions in nltk (natural language toolkit) where it extracts all the nouns from the documents thereby creating its synsets. Each noun may have different synsets. So, each noun is replaced by the most appropriate sense by finding the similarity between the senses. The similarity measure that is used here to find the similarity between senses is a combination of Wu-Palmer and Banerjee Pederson algorithm.

$$\delta(c_p, c_q) = \frac{2d + S}{L_p + L_q + 2d + S} \tag{1}$$

Let N= $\{n_1, n_2, ..., n_p\}$ which denotes the set of all nouns in a document d, where ni £ N. Let C_i =$\{$ $ci_1, ci_2, ..., ci_l\}$ denotes the set of senses corresponding to each noun in the document according to wordnet ontology. To determine the accurate sense of a noun n_i is by computing the sum of similarity to other noun senses in d by the following measure.

$$c_i = max_{c_{il} \epsilon c_i} \sum_{n_{ij} \epsilon d} max_{c_{jm} \epsilon c_j} S(c_p, c_q) \tag{2}$$

It will find the most accurate sense and replace the noun with that sense in the original document.

B. Construction of lexical chains

Lexical chains are implemented here to extract the core semantic features of a text. It is built on the basis of disambiguated semantic concepts. Let RN= $\{identity, synonym, hypernym, meronym\}$ be a set of semantic relations that may exist between the senses and R= $\{r_1, r_2 = r_1, r_3, r_4\}$ corresponding to the weight of RN. Identity and synonym are considered as one relation because all the nouns in the text have been replaced by the most appropriate sense using Wordnet. Each lexical chain may have one or more nodes and each node represent sense of a word and each link can represent a relation. (identity, synonym, hypernym, meronym) between the senses. To extract the core semantics one or more nodes and each node represent sense of a word and each link can represent a relation of a given document, the semantic importance of word sense must be evaluated initially. Let N = $\{n_1, n_2, ..., n_p\}$ be the set of nouns in a document d and let F= $\{f_1, f_2, , \{f_1, f_2, ..f_p\}$ be the corresponding frequency of occurrence of nouns in d. Let C = $\{c_1, c_2, ..., c_q\}$ be the set of disambiguated concepts that corresponding to N. Given a document d, a set of nouns N, a set of frequencies F and a set of concepts C, let W= $\{w_1, w_2, ..., w_q\}$ as the set of corresponding weight of disambiguated concepts in C, if c_i (c_i ϵC) $is\ mapped\ from$ n_k and n_m (n_k, n_m ϵN), $then\ the\ weight\ of$ c_i is computed by

$$w_i = f_k + f_m \tag{3}$$

1) Score of a concept:

Let C = $\{c_1, c_2, ..., c_q\}$ be the set of disambiguated concepts(word senses), and let W=$\{w_1, w_2, ..., w_q/$ be the set of corresponding weight of disambiguated concepts in C. Let RN = $\{identity, synonym, hypernym, meronym\}$ be the set of semantic relations, and let R = $\{r_1, r_2 = r_1, r_3, r_4\}$ be the set of the corresponding weight of relation in RN. Then the score of a concept c_i($c_i \epsilon C$) $in\ a\ lexical\ chain\ is\ computed\ by$

$$S(c_p, c_q) = w_i \times r_i + \sum \{w_p \times H(c_p, c_q, k) \times r_k \tag{4}$$

where,

$$H(c_i, c_p, k) = \begin{cases} 1 & if there is an edge of RN_k between c_i and c_p \\ 0 & otherwise \end{cases} \tag{5}$$

A large value of S(c_i) indicates that c_i is a semantically important concept in a document. The relation weight r (r e R) depending on

the kind of semantic relationship and it is in the order listed: identity, synonym, hypernym (hyponym), meronym (thus, $r_1 = r_2 > r_3 > r_4$).

2) Score of a lexical chain:

The score of each lexical chain is calculated by adding the score of each concept in that chain. Let L = $\{L_1, L_2, ..., L_m\}$ be a set of lexical chains of a given document, Li e L. Let ci = $\{c_{i1}, c_{i2}, ..., c_{iq}\}$be a set of disambiguated concepts in L_i.$LetS(c_{il})$ be the score of concept c_{il} (c_{il} e c_i). Then, the score S(L_i) of lexical chain in a document is defined as

$$S(Li) = \sum_{l=1}^{q} S(C_{il}) \tag{6}$$

The lexical chain which is having highest score will represent the theme of that particular document.

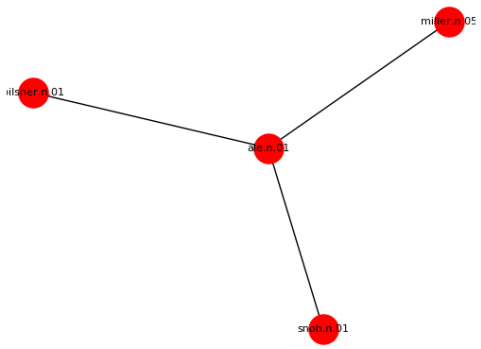
C. Similarity Checking

Here we have a set of documents with Lexical chains. For each document d, the lexical chain having higher score is taken and the nodes are assigned to a list. These lists will be added to a multidimensional list which is a global list. To find out the similarity between the documents we need to find the common senses present in both document. This is carried out by taking the intersection of the lists which represent the common senses between the documents. If the result of the intersection is null, it means there is no similarity between the documents.

II. Results and Analysis

Construction of lexical chains will help us to find out the theme of each document when there are thousands of documents. Analyzing each documents lexical chain which has the highest score will give the theme of its core content. To build lexical chains, word sense disambiguation must be done first. Word sense disambiguation will replace all the nouns in the document with the most appropriate sense according to its content. Example:
text = "I like beer. Miller just launched. A new pilsner. but, because I am a beer snob, I am only going to drink pretentious Belgian ale."

The synsets of nouns formed from the above text are:
Synset ('pilsner.n.01'), Synset ('beer.n.01'), Synset ('snob.n.01'), Synset ('ale.n.01')
Lexical chains contain sequence of words which has some semantic relations. In a lexical chain, vertices represent senses and edges represent the relation between them. The relation between the senses is either hypernym or meronym. The senses which satisfy this kind of relation will form lexical chains. For each lexical chain, there is a Master node. The other senses/nodes are added to this chain according to the semantic relation of those senses with the master node. Fig(i) represents the lexical chain corresponding to above mentioned text.



The score of a lexical chain is the sum of the score of the concepts. The score of the concept can be find using the weight of each concept. The weight of a concept is nothing but the count of the noun which replaces the particular concept.

Concept score of ale can be find out by:-
ConceptScoreOne=conceptweightOne1+
conceptweightcountsTwo$\times 1 \times r3$.
Concept score of beer can be find out by:-
ConceptScoreTwo=conceptweightTwo1+
conceptweightcountsOne.$\times 1 \times r3$.

Lexical chain score is found out by:-
Lexicalchainscore=Lexicalchainscore+Lexical.
Concept count of ale and beer are:-
ConceptCount of ale.n.01
1
ConceptCount of beer.n.01
1

Concept score of ale and beer are:-
(u'ConceptScore of ale.n.01', 1.25)
(u'ConceptScore of beer.n.01', 1.25)

lexical chains score:- 2.5

III. Conclusion

Lexical chains enable to identify the basic theme of a document. For that, here we are using word Sense disambiguation, a modified WordNet based semantic similarity measurement and lexical chains to extract the core semantic features. Intersection of the word senses in lexical chains will give the similarity. However, the limitation in our research is: i) In WordNet lexicon, certain important words are not included so that it will not consider those concepts while evaluating the similarity condition. ii) Using large documents increase the no of nodes in a lexical chain which increase the complexity of the graph.

IV. Future Work

We would like to cluster the documents having same context into a single cluster such that each cluster will represent a set of documents which have same context.

References

[1] 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*, vol. 42, no. 4, pp. 2264–2275, 2015.

[2] S. Banerjee and T. Pedersen, "An adapted lesk algorithm for word sense disambiguation using wordnet," in *International conference on intelligent text processing and computational linguistics*. Springer, 2002, pp. 136–145.

[3] D. Jayarajan, D. Deodhare, and B. Ravindran, "Document clustering using lexical chains," 2007.

[4] M. Jarmasz and S. Szpakowicz, "Not as easy as it seems: Automating the construction of lexical chains using rogets thesaurus," in *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer, 2003, pp. 544–549.

[5] A. Ghose and B. Ravindran, "Supervised lexical chaining," Ph.D. dissertation, Masters thesis, Indian Institute of Technology, Madras, 2011.

[6] M. Augat and M. Ladlow, "Cs65: An nltk package for lexical-chain based word sense disambiguation," *Word J. Int. Linguist. Assoc*, 2004.

[7] B.-Y. Kang and S.-J. Lee, "Document indexing: a concept-based approach to term weight estimation," *Information processing & management*, vol. 41, no. 5, pp. 1065–1080, 2005.

[8] B.-Y. Kang, D.-W. Kim, and S.-J. Lee, "Exploiting concept clusters for content-based information retrieval," *Information sciences*, vol. 170, no. 2-4, pp. 443–462, 2005.

[9] V. Nastase and S. Szpakowicz, "Word sense disambiguation in roget's thesaurus using wordnet," in *Proc. of the NAACL WordNet and Other Lexical Resources Workshop. Pittsburgh*, 2001.

[10] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188–1196.

[11] M. Granik and V. Mesyura, "Fake news detection using naive bayes classifier," in *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. IEEE, 2017, pp. 900–903.

[12] S. Aphiwongsophon and P. Chongstitvatana, "Detecting fake news with machine learning method," in *2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 2018, pp. 528–531.

[13] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.