Shruti Gupta (A20381966) Sagar Mane (A20379756)

HW3 Lexical Chain

Lexical Chain Evaluation

Steps:

i) Read text file

ii) Preprocess text file by removing number, stop words, punctuation and non-alpha numeric character. For more accurate result we also lemmatize and stem the word to its based form. After preprocessing sentences in text file, we are creating list of words as tokens.

iii) Find the similarity between each word in tokens by using ***wordnet synsets***. We are using Wu-Palmer similarity and Jiang-Conrath similarity to get more accurate similarity between words and generating lexical chain. Using list of words in tokens we find frequency of each words in text files.
Note: Threshold value for Wu-Palmer similarity is 0.6 while for Jiang-Conrath is 0.08

Output:
List of Tokens:

```
['girl',
 'women',
 'person',
 'hat',
 'skirt',
 'girl',
 'shirt',
 'skirt',
 'skirt',
 'girl',
 'women',
 'person',
 'shirt',
 'rome',
 'mumbai',
 'one',
 'two',
 'ten',
 'one',
 'ten',
 'three',
 'want',
 'told',
 'woman']
```

Lexical Chain with all POS:

```
Chain 1: person(2), girl(3), women(2), woman(1),
Chain 2: hat(1), shirt(2), skirt(3),
Chain 3: rome(1), mumbai(1),
Chain 4: three(1), two(1), ten(2), one(2),
Chain 5: want(1),
Chain 6: told(1),
```

Lexical Chain with only Noun:

```
Chain 1: person(2), girl(3), women(2), woman(1),
Chain 2: hat(1), shirt(2), skirt(3),
Chain 3: rome(1), mumbai(1),
Chain 4: three(1), two(1), ten(2), one(2),
```

Observation/Conclusion:

We observed that after preprocessing text file, we generate list of tokens. Tokens contains four types of words like person, cloth, city and number. Using ***wornet*** we generate lexical chain and each lexical chain has similar types of words. Chain 1 contains all list of words which are related to person. Chain 2 contains all list of words which are related to clothes. Chain 3 contains all list of words which are related to cities. Chain 4 contains all list of words which are related to number.

The methods like jcn similarity uses IC(Information Content) to measure the similarity of different word senses. The conventional way of measuring the IC of word senses is to combine knowledge of their hierarchical structure from an ontology like WordNet with statistics on their actual usage in text as derived from a large corpus
Thus, we have used the ic-brown.dat file lists every word existing in the Brown corpus and their information content values (which are associated with word frequencies) that will in turn combine the power of hierarchical word sense with that from a large corpus of brown(ic-brown.dat).
In addition, we can also use other semantic measures like Resnik and Lin which are in turn again based on the IC measure to find the similarity of the word senses apart from what has been used.
The semantic measures by jcn require the use of a corpus in addition to the WordNet. These measures combine WordNet with corpus statistics and they are shown to achieve better correlations to human judgment than using WordNet alone.

How to further improve result:

There are several ways to improve the results or provide more cross-validation. Some of the proposed approaches include:

Latent Symantic Analysis (LSA): LSA find the relation between words in texts. This method starts from the term-doc matrix computed on the corpus segmented into chunks and then applies a singular value decomposition in order to compute the most important singular values. Then, it produces a latent semantic space, which is used to compute similarities between different words.

Latent Dirichlet Allocation (LDA): LDA extract topics from text. In LDA, documents are represented as random mixtures of latent topics, where each topic is characterized by a set of pairs word-probability, representing the probability that a word belongs to topic.

Shruti Gupta (A20381966) Sagar Mane (A20379756)

Extra Credit:

Extra credit: Use the lexical chains to automatically create a summary of the input article.

Implementation/Steps:

I. Find score of each lexical chain
II. Find score of each sentence using score of lexical chain
III. Based on sentence score sort the sentences
IV. Display top sentence as summary

Note: We are displaying only top 5 sentences as summary.

Input File:

```
Thirty-eight. That's another number I want you to remember.

My second year in the league, we won 38 games. I played in all 82 games that year and got
a lot more minutes. My numbers improved to eight points and 10 rebounds a game. And most
important (!) I wasn't sitting on the floor next to the bench anymore.

We were moving in the right direction, and I think there was one main reason why: We got
Quin. It was Coach Snyder's first year as our coach.

Quin really surprised me the first time we met. It was in training camp in September 2014.
I had just spent the summer playing in the World Cup for the French national team. To be
honest, I didn't think Coach knew who I was. One of the first conversations we had, he
came up to me and told me he had watched every game France played in the World Cup.

It wasn't b.s. — he was bringing up specific plays from specific games. He remembered how
we lost to Spain in the group stage — by 24 points — and he remembered how we faced them
again in the quarterfinals and won. We surprised a lot of people. Spain had the Gasol
brothers, Serge Ibaka, Ricky Rubio, a great team. It was considered one of the best
European teams. France was kind of forgotten about that year.
```

Output/Summary:

```
['It was Coach Snyder's first year as our coach.',
 'One of the first conversations we had, he came up to me and told me he had watched every game France played in the
World Cup.',
 'He remembered how we lost to Spain in the group stage — by 24 points — and he remembered how we faced them again in
the quarterfinals and won.',
 'Spain had the Gasol brothers, Serge Ibaka, Ricky Rubio, a great team.',
 'I had just spent the summer playing in the World Cup for the French national team.']
```