Advanced Machine Learning Project

INSTRUCTOR :

JIE TANG

ASSOCIATE PROFESSOR

JUN ZHU

ASSOCIATE PROFESSOR

Article Generator

KAZI ABIR ADNAN (安奈威)
2014280162
LEUNG WAI KIM
2014270019

Introduction

- A standalone software to generate article on a given topic

Example

- Say, you want to create an article about Cristiano Ronaldo

Input

Output

Cristiano Ronaldo dos Santos Aveiro, GOIH, (born 5 February 1985), known as Cristiano Ronaldo (Portuguese pronunciation: [kɾiʃtiˈɐnu ʁuˈnaldu]), is a Portuguese professional footballer who plays for Spanish club Real Madrid and the Portugal national team. Ronaldo later claimed that he lived in poverty, sharing a room with his brother and sisters. Ronaldo later said of the incident, "He disrespected me". At age 14, Ronaldo agreed with his mother to focus entirely on football.

At the age of 15 Ronaldo was diagnosed with a racing heart, a condition that might have forced him to give up playing football. Of a possible move to Liverpool, he said "Liverpool are one of the best clubs in England and it would be a dream for any player to represent a club of such traditions". However he came to the attention of Manchester United manager Alex Ferguson in the summer of 2003, when Sporting defeated United 3–1 in the inauguration of the Estádio José Alvalade in Lisbon.

Ronaldo's performance impressed the Manchester United players, who urged Ferguson to sign him. In April 2013, Sporting honoured Ronaldo by selecting him to become the 100, 000th member of the club. After the 2006 FIFA World Cup, in which he was involved in an incident where club teammate Wayne Rooney was sent off, Ronaldo was linked with moves away from United, with Spanish clubs Real Madrid and Valencia

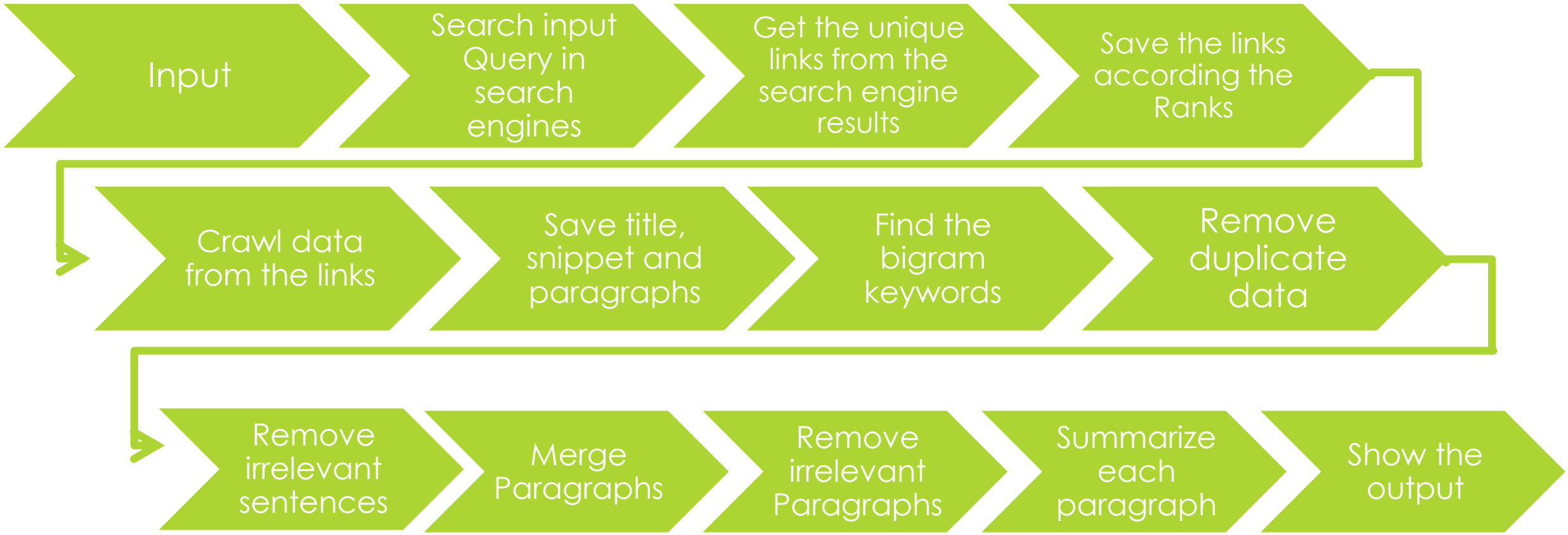
Right now what we do

- Open an editor
- Write something on our own
- Search on internet and get relevant document
- Use relevant information from the document
- Add references of that information
- Rearrange or create paragraphs on subtopics

Our solution

- Automate this process of article creation
- We call it Article Generator

High level design



Dataset

- Our dataset is the search result links of a query topic
- We have used 3 search Engines to find the links relevant with query topic
  - Google, Bing and Yahoo
- Crawled the title, snippet and the link addresses of each search engine results
- Crawled the paragraphs of each saved links
- Emphasized on highest ranked documents

Methods

- Our system has 6 main sub process
  1. **Preprocessing step** : Removing duplicate paragraphs
  2. **Keyword Finding** : Find the keywords from whole corpus
  3. **Sentence Matching** : Make new paragraph by sentence clustering
  4. **Remove irrelevant and duplicate sentences** : Remove paragraphs that has irrelevant sentences
  5. **Merge Paragraphs** : Merge the similar paragraphs from the previous step
  6. **Summarization** : Summarize each of the paragraphs remaining important lines

1. Removing duplicate paragraph

- We have used Cosine Similarity measure to find similarities between raw input paragraphs from different links (Threshold : 85%)
  - Frequency of words as input vector

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

- N.B : First We tried edit distance. But the result was very poor. Because the length of the paragraph varies a lot and similarity score for this was always low.

2. Finding the keywords

- Frequency of words is the key to find keywords
- Preprocessed each word from every sentences. Like making lowercase, removing illegal characters, Stemming (KStem), Stopwords filtering etc.
- Check every combination of two words to find bigram Keywords.
- We can try it for ngram keywords. But, It will make the process very slow.
- N.B :
  - For stemming, Porter Stemmer algorithm is very hard stemmer to analyze words. That is why we have used KStem algorithm.
  - We have used Apache Lucene token analyzers. Apache Lucene project has excellent Natural language processing library.
- Used top 20 frequent words as our keywords

3. Make new paragraphs using sentence clustering

- Intermediate step to remove some of the lines from input paragraphs
- Why Sentence Clustering?
  - Basically, our temporary target is to find similar sentences. So, that we can find the irrelevant and duplicate ones
  - It will make a cluster of irrelevant sentences (ex. Footer words, web links, email addresses, thank you words or common blog sentences etc). So, I can remove this irrelevant sentences from final article.
- **Group Average Agglomerative Clustering to cluster sentences
  - It is hard clustering and starts with each sentence as a cluster.

$$\text{SIM-GA}(\omega_i, \omega_j) = \frac{1}{(N_i + N_j)(N_i + N_j - 1)} \sum_{d_m \in \omega_i \cup \omega_j} \sum_{d_n \in \omega_i \cup \omega_j, d_n \neq d_m} \vec{d}_m \cdot \vec{d}_n$$

  - All pair similarities between sentences.
  - Core idea is objects being more related to nearby objects
  - Merge with each other at certain stages.(merge until similarity reaches threshold)

** Ref : Web document clustering: a feasibility demonstration
**http://nlp.stanford.edu/IR-book/html/htmledition/group-average-agglomerative-clustering-1.html

Example

I hope you found this post interesting.

My name is Vasilis Vryniotis.

Learn more.

Your email address will not be published.

Required fields are marked *.

How to build your own Twitter Sentiment Analysis Tool. The importance of Neutral Class in Sentiment Analysis.

Subscribe to our newsletter and get our latest news! It is fast and enables us to integrate out some variables while sampling another variable.

2013-2015 © Datumbox.

All Rights Reserved.

Privacy Policy | Terms of Use.

Stay tuned for the upcoming articles!

4. Remove irrelevant Clusters

- Now, We have clusters of similar sentences.
- Remove the clusters which doesn't have any keywords. That means, a cluster which doesn't contain any relevant information to our topic
- We have used a threshold here. Like, if (0-3) keywords are present in the cluster, then remove it.

Example

Choose from six languages and in 13 regions worldwide. Mohammad Raza, Sumit Gulwani, and Natasa Milic-Frayling, Compositional Program Synthesis from Natural Language and Examples, March 2015. Asta Roseway, Yuliya Lutchyn, Paul Johns, Elizabeth Mynatt, and Mary Czerwinski, BioCrystal: An Ambient Tool for Emotion and Communication , in IJMHCI, March 2015. Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig, Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding, in IEEE/ACM Transactions on Audio, Speech, and Language Processing, IEEE – Institute of Electrical and Electronics Engineers, March 2015. Bin Gao and Tie-Yan Liu, Global Optimization for Advertisement Selection in Sponsored Search, in JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY, 1 March 2015. Svore, Quantum Nearest-neighbor Algorithms for Machine Learning, in Quantum Information and Computation, vol. 15, no. 3&4, pp. 0318-0358, Rinton Press, March 2015. The launch point for more sophisticated methods, like random forests and boosting. Nanodegree Credentials Georgia Tech Program Udacity for Business Udacity for Veterans Help and FAQ Feedback Program. Lecture slides for instructors, in both postscript and latex source. Errata for printings one and two (postscript)(pdf).

Choose from six languages and in 13 regions worldwide. Mohammad Raza, Sumit Gulwani, and Natasa Milic-Frayling, Compositional Program Synthesis from Natural Language and Examples, March 2015. Asta Roseway, Yuliya Lutchyn, Paul Johns, Elizabeth Mynatt, and Mary Czerwinski, BioCrystal: An Ambient Tool for Emotion and Communication , in IJMHCI, March 2015. Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig, Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding, in IEEE/ACM Transactions on Audio, Speech, and Language Processing, IEEE – Institute of Electrical and Electronics Engineers, March 2015. Bin Gao and Tie-Yan Liu, Global Optimization for Advertisement Selection in Sponsored Search, in JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY, 1 March 2015. Svore, Quantum Nearest-neighbor Algorithms for Machine Learning, in Quantum Information and Computation, vol. 15, no. 3&4, pp. 0318-0358, Rinton Press, March 2015. The launch point for more sophisticated methods, like random forests and boosting. Nanodegree Credentials Georgia Tech Program Udacity for Business Udacity for Veterans Help and FAQ Feedback Program. Lecture slides for instructors, in both postscript and latex source. Errata for printings one and two (postscript)(pdf).

5. Merge similar paragraphs

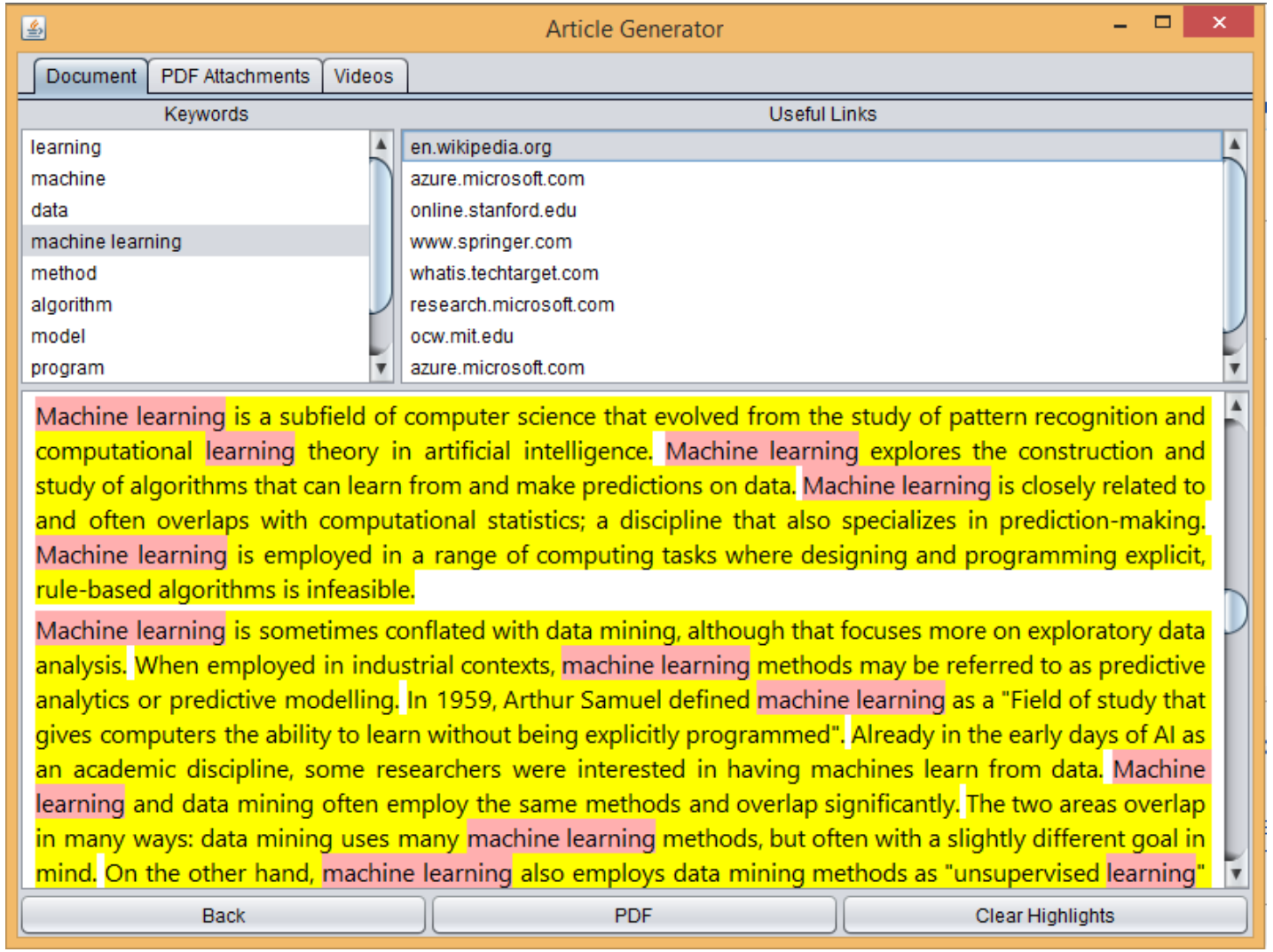
- Now the next step is to merge the paragraphs
- We have used “K Means” algorithm here to merge the paragraphs
  - Centroid based clustering
  - Fixed K
  - TF-IDF as feature vector
  - Just merge the similar paragraphs
- Here input data is the varied number of paragraphs after processed by GAAC.
- Similarity between paragraphs using Cosine distance.
- Here K Means clustering is customized by sorting each paragraphs of cluster by ranking.
  - Highest ranked document’s information is highly preferred

6. Summarization

- Goal is to summarize each paragraph.
- We have used here Extract-based summarization.
- Find the significant sentences in the paragraph.
  - Give score to each sentence. Scoring is done by keywords of whole corpus and significant words in each paragraph.
  - If sentence contains keyword, score is added by highest point.
  - Significant word is highest frequency words of the paragraph.
  - Stemming and all kind of filtering is done here also during score calculation.
- We have a compression ratio (like 40%) to calculate the length of summary
- Then we sort the scores of each sentence and choose the highest ones.
- We do not reorder the sentences here. Just, picking the important lines.

Result

- An article about Machine Learning
- Shows the keywords
- Reference links
- PDF creation option
- Highlight document
- Show attachments



Application & Tools

- Desktop application
  - Right now too heavy to create a webapp
  - IP block as crawling search engine results
- Java
- External libraries
  - Jsoup – to crawl data
  - Json parsers
  - Text to pdf converter
  - Apache Lucene – NLP analyzer
  - Word2Vec, Sentence2Vec library

System Features

- Highlight keywords
- Highlight sentences by web pages
- Show the original link document in a separate view
- Rearrange the paragraphs manually
- Export as a pdf file
- PDF and video attachments
- Spell checker before searching the query topic on search engine

Comparison with existing system

- There are some online composition generator
- Some of those systems just crawls the paragraphs from the link and let users to create an article by choosing from those paragraphs
- Most of these software's are not free.

Analysis

- First try was only Sentence clustering. But, It just clusters all the sentences together. Example:
  - **The Dirichlet process can** also be seen as the infinite-dimensional generalization of the Dirichlet distribution. **The Dirichlet process can** be used to circumvent infinite computational requirements as described above. **The Dirichlet Process can** also be used for nonparametric hypothesis testing
- Still, the result of this process is not satisfactory. Because,
- Transitional words:
  - Transitional words like accordingly, because of, consequently, thus etc should be handled as a special case. This words has dependency with previous sentence.
- Paragraph flow:
  - To make a meaningful article we have to maintain the flow of the paragraph. Like Introduction, Description, Conclusion etc.
  - Paragraphs might have dependency with the previous one. So, we have to maintain the dependency.
- Inconsistent generation of article
- Very slow if input paragraph number is large

Improvement

- Changing the features can improve it?
  - Previously we have used TF-IDF as our feature vector
  - Probability can not extract the meaning of the whole paragraph
- Use Sentence2Vec (extend from Word2Vec) to generate the feature of the sentences
- Run K-means cluster (feature size of sentence2vec is less than TF-IDF)
- We believe the generated vector of the sentence is more representative of its meaning

Word2Vec

- Unsupervised algorithm for learning the meaning behind words
- Efficient implementation of the continuous bag-of-words and skip-gram architectures for computing **vector representations** of words.

Word	Cosine distance
spain	0.678515
belgium	0.665923
netherlands	0.652428
italy	0.633130
switzerland	0.622323
luxembourg	0.610033
portugal	0.577154
russia	0.571507
germany	0.563291
catalonia	0.534176

- The words similar to the word “france” :

- Interesting properties of the word vectors:
  - *vector('Paris') - vector('France') + vector('Italy') ~ vector('Rome')*
  - *vector('king') - vector('man') + vector('woman') ~ vector('queen')*

Word2Vec Model

- skip-gram model.
- The training objective of skip-gram is to learn word vector representations that are good at predicting its context in the same sentence.
- Mathematically, given a sequence of training words *w*1,*w*2,…,*wT*, the objective of the skip-gram model is to maximize the average log-likelihood

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j=-k}^{j=k}\log p(w_{t+j}|w_t)$$

where *k* is the size of the training window.

Word2Vec Model (cont)

- In the skip-gram model,
  every word w is associated with two vectors u_w and v_w which are vector representations of w as word and context respectively.
- The probability of correctly predicting word w_i given word w_j is determined by the softmax model, which is

$$p(w_i|w_j) = \frac{\exp(u_{w_i}^\top v_{w_j})}{\sum_{l=1}^{V} \exp(u_l^\top v_{w_j})}$$

  where V is the vocabulary size

- Softmax Model complexity: $\log p(w_i \mid w_j)$
  - proportional to V, which can be easily in order of millions.
- Hierarchical softmax, which reduced the complexity to $O(\log(V))$

Sentence2Vec / Paragraph2Vec

- Modify the algorithm of the word2Vec **unsupervised learning of continuous representations for larger blocks of text**, such as sentences, paragraphs or entire documents.
- Use a vector to represent the sentence

Quoc Le, Tomas Mikolov, "Distributed Representations of Sentences and Documents"

Future perspective

- Can be used as a wiki page generator for a topic
- Can be used in social search engine to create automatic page(ex. Facebook)
- Can help bloggers to suggest article paragraphs before writing (Like searching suggestion)
- Can be used to solve assignments ! ! !

Conclusion

- Related to natural language processing
- Not that easy to implement it only by clustering
- Though our system has shortcomings, it can help users to create an article about a topic

Thank you

CLICK TO START DEMO