CS 7337 – Natural Language Processing

Final Exam

Instructions: Clarity of answers is more important than length of answers. Although not required (unless indicated otherwise), feel free to use graphs, charts, visuals, et al in your answers if you feel these artifacts can help support your answers. There are no bonus points for using these artifacts. Submit your answers in PDF or Word document format.

Due date: See course wall announcement.

Q1.

1. [5 pts] What is Distributional Hypothesis in the context of distributional semantics? Give a short explanation with some examples.

Solution:

Distributional hypothesis in distributional semantics is that the meaning of words that occur in the same context tend to have similar meanings.

Example: The meaning of the words “wonderful” and “splendid” are similar as they can be exchanged in the same context.

Source: <https://aclweb.org/aclwiki/Distributional_Hypothesis>, <https://pdfs.semanticscholar.org/3ff9/b56229c04a4426e532b37d26f02610c533fb.pdf>

1. [5 pts] Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) are two widely used techniques for topic modeling. Give a short overview of the two approaches and any similarities/differences between them.

Solution: Both LSA and LDA are ways to implement organic topic modeling.

Latent Semantic Analysis (LSA): LSA-based topic modeling tries to find groups of words associated with the largest variances between documents in the corpus.

Latent Dirichlet Allocation (LDA): LDA construes topics as groups of words that have high cooccurrences among different documents in the corpus.

Comparison:

|  |  |  |
| --- | --- | --- |
| **Comparison Parameters** | **LSA** | **LDA** |
| **Tends to separate word senses?** | No | Yes |
| **Time to train** | ½ x | 1x |
| **Time to execute** | 2x | 1x |
| **Typical Number of topics for optimal results** | 300-500 | 30-50 |
| **Model** | Singular Value Decomposition | Generative probabilistic model |
| **Calculations** | Topic-topic matrix | Probabilistic distributions over words |

Source: <https://towardsdatascience.com/2-latent-methods-for-dimension-reduction-and-topic-modeling-20ff6d7d547>, Unit 13 Lecture Notes

Q2.

* 1. [5 pts] You are a Data Scientist for an e-commerce site for electronics which also supports 3rd party sellers. You would like to build a system to find and match the same products that sellers on your website sell so that you can present them in a single product page. You decide to use product titles to compute product similarity. Which similarity metric, Jaccard or Cosine, would you use and why?

Solution: For comparison of product titles, I would recommend using Jaccard similarity metric. Between the Jaccard and Cosine similarity metrices, Jaccard similarity scores considers only unique sets of words for the product topics. By using the Jaccard similarity scores, we ensure that repeating words do not reduce their similarity.

Source: <https://towardsdatascience.com/overview-of-text-similarity-metrics-3397c4601f50>

* 1. Consider the following table which lists electronic items for sale on two ecommerce shopping websites. Products in row -1 are the same product, row-2 are different TV models of the same brand and row-3 are different products.

|  |  |
| --- | --- |
| Product Title 1 (Site 1) | Product Title 2 (Site 2) |
| 50 Inch Class H6570G 4K Ultra HD  Android Smart TV with Alexa  Compatibility 2.5” 2020 Model Black  Silver White HDR LED | Hisense H6570G |
| QN75Q90TAFXZA crystal 2.5” Quantum  LCD | Samsung crystal UN55TU8000FXZA QLED |
| EGLF2 50 Ultra Full Motion Articulating  TV Wall Mount Bracket swivel full | VIZIO EGLF2 |

[10 pts] Considering your answer to 2a) will your similarity calculation approach work on this dataset? Explain with examples.

Solution: Let’s calculate the Cosine Similarity Score and Jaccard Similarity Score by hand.

To calculate the Cosine Similarity, we follow the below steps:

1. Calculate the term frequency using bag of words
2. Normalize the term frequencies with the respective magnitudes
   1. Calculate the square root of sum of squares – This Is the normalized value.
   2. Normalize by dividing the term frequency (Step 1) by norms (Step 2a)
3. Calculate the Cosine Similarity Score by getting dot product of the 2 vectors.

To calculate the Jaccard Similarity, we follow the below steps:

1. It is recommended to perform lemmatization before we proceed
2. Get the count of distinct words that occur in both the vectors
3. Divide it by the total number of distinct words.

Based on the above steps, the calculated scores are mentioned below:

|  |  |  |
| --- | --- | --- |
| Product | Cosine Similarity Score | Jaccard Similarity Score |
| Product 1 (Same product) | 0.1543 | 0.0417 |
| Product 2 (Same brand diff product) | 0.25 | 0.1 |
| Product 3 (Different product) | 0.189 | 0.0714 |

Inference: Based on the other table, it is clear that both the methods do not identify the Product 1 correctly. In both cases, the score for Product 1 is the lowest. For Product 2 and Product 3, the score trends are on expected lines. Overall, in the calculation of the Jaccard and Cosine similarity scores, the number of common words and total number of distinct words had a bearing on the final score. All 3 products had only 1 word in common. Eventually, the length of the text of the product titles had a bearing on the final score. Overall, the approach chosen in 2a will not work for this dataset.

Source: <https://towardsdatascience.com/overview-of-text-similarity-metrics-3397c4601f50>

[10 pts] Suppose that you are given IDF scores for all tokens (see Table below). Can this help you come up with a better approach for computing title similarity? Explain with examples.

|  |  |
| --- | --- |
| Product Title 1 (Site 1) | Product Title 2 (Site 2) |
| 50(6.3) Inch (8) Class (8.5) H6570G (10.2) 4K (9.4) Ultra (6.6) HD (5.7) Android (2.6) Smart (6.1) TV (3.9) with (4) Alexa (6.9) Compatibility (15.6) 2.5” (5.7) 2020 (6.8) Model (12.6) Black (6.8) Silver (7.8) White (12.6) HDR (12.2) LED (6.9) | Hisense(9.5) H6570G(10.2) |
| QN75Q90TAFXZA(13.7) crystal(11.3)  2.5”(5.7) Quantum(7.8) LCD(6.8) | Samsung(8) crystal(11.3) UN55TU8000FXZA(16.5) QLED(4) |
| EGLF2(15.6) 50(6.3) Ultra(6.6) Full(5.6)  Motion(6.7) Articulating(2.6) TV(3.7) Wall(8.5) Mount(9.5) Bracket(11)  swivel (8.5) full (5.6) | VIZIO(10) EGLF2(15.6) |

Solution: Let’s calculate the Similarity Score using tf-idf by hand.

To calculate the Similarity using tf-idf, we follow the same steps mentioned in the previous answer for Cosine Similarity using bag of words method but substitute the tf-idf values instead of using the term frequencies (TF).

|  |  |
| --- | --- |
| Product | Cosine Similarity Score |
| Product 1 (Same product) | 0.1921 |
| Product 2 (Same brand diff product) | 0.2835 |
| Product 3 (Different product) | 0.4705 |

Inference: Here again, the cosine similarity seems to increase between products as similarity decreases. This is the inverse of the expected property. This could be attributed to the mismatch in the text length between the 2 sites. Ideally, the product titles would be of similar length and the cosine similarity or the Jaccard similarity could be used to compare them and identify similar products. For future analysis, we could also explore removing unimportant words from Site 1 Product titles. For example, the color (Silver, black, white) does not add any information to the product.

Q3.

1. [10 pts] Recommender systems are a subtype of information filtering systems that help users discover new and relevant items by presenting items similar to their previous interactions or preferences. Some famous examples of recommender systems are Amazon’s “Books you may like” and Netflix’s “Because you watched” carousels.

You are building a recommender system for your food delivery service startup and have data on co-purchases for food items f1, f2, . . ., fn (for example, food item f1 is commonly bought together with food item f4). How can you use techniques such as Word2Vec to recommend similar items to users who may have bought or show interest in any one of the items?

Solution: Recommender systems are algorithms aimed at suggesting relevant items to users. Recommender systems are popular across many industries like e-commerce (Amazon, Yelp, Uber Eats, etc.), Streaming Services (Netflix, Hulu, Amazon Prime, etc.), Online Advertisements (Google, Bing, etc.), and many more.

Word2Vec uses a neural network model to learn word associations from a large corpus of text. Applications of Word2Vec usually includes detecting synonyms words and suggesting missing words of a partial sentence.

In our recommender system, we would typically train the previously placed orders to form the corpus. The model would convert each order item into a vector. Then, we would curate a set of tags and convert them into vectors as well. After this, we would be to examine each order item vector and compare with the tag vector. This will help us classify the order item as the tag that was most similar. Finally, we validate the classification results using cluster visualization and deploy the recommender system to the app. The app displays the recommendation based on the tag of the items chosen in the cart.

Source: <https://developer.squareup.com/blog/caviars-word2vec-tagging-for-menu-item-recommendations/> <https://www.analyticsvidhya.com/blog/2019/07/how-to-build-recommendation-system-word2vec-python/>

1. [10 pts] Word2Vec implements two different neural models: skip-gram and continuous bag of words (CBOW). Briefly explain the differences between the two models. Under which circumstances would you prefer the skip-gram model over CBOW?

Solution: Both the Continuous Bag of Words and Skip-gram models learn the underlying word representations for each word using neural networks.

Continuous Bag for Words predicts the target word by leveraging all words in its neighborhood. The CBOW uses a pre-defined window size surrounding the target word to create a context vector in order to predict the target word. Each word is coded in one-hot form. The training objective is to maximize the conditional probability of observing the actual output word given the input context words, with regard to the weights. CBOW trains several times faster compared to the skip-gram model and performs slightly better for frequently occurring words.

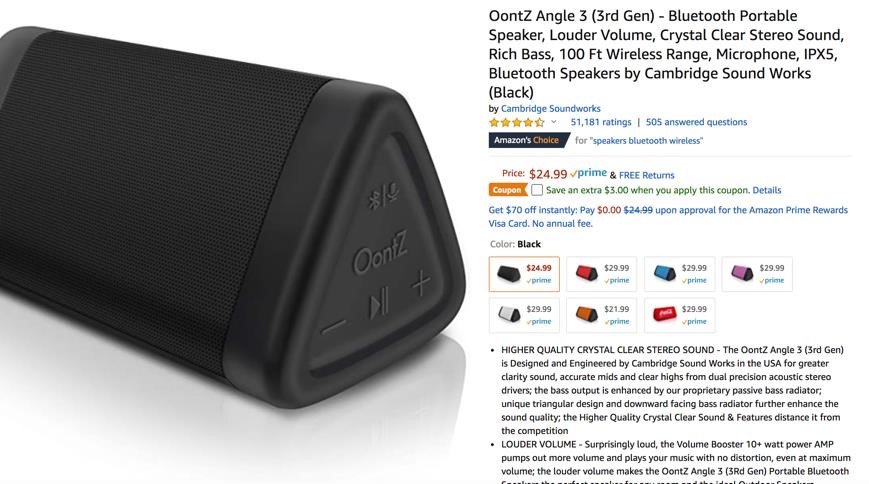
In the Skip-gram model, the components need not be consecutive in the text under consideration. Skip-gram works well with sparse data and handles rare words and phrases admirably. The training objective of the skip-gram model is to minimize the summed prediction error across all context words in the output layer. The focus is the single input vector and the target context of words are the output layer.

Circumstances we would prefer skip-gram model over CBOW: Skip-gram works well with small amount of the training data and represents well even in rare words or phrases.

Source: <https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314>, Class Slides Week 10.

Q4.

You are building a product classification system for an online electronics store. The system should classify an incoming stream of millions of products to one of the 3000+ leaf level product types in the taxonomy such as laptops, smart TVs, wireless headphones, car speakers, among others. The system should be very precise because it’s important to assign products to the right category to facilitate the customer shopping experience. Each instance in your dataset has product title, description and image fields. See example below:



1. [5 pts] What features would you use for your machine learning-based classifier?

Solution: We would use Multi-class classification to address the machine learning-based problem. I would propose using Chimera: Large-scale classification using Machine learning, rules, and crowd sourcing.

Source: <https://blog.acolyer.org/2016/01/28/chimera/>, Class Slides Week 12.

1. [5 pts] Assume that you only have access to product titles in your dataset (i.e., you have less data to play with) instead of product titles, description and images. How will this affect feature engineering and the NLP pipeline for your classifier?

Solution: Working with just the product title is better for the classification since the product description would vary from vendor to vendor. The length of each instance is similar, and the titles are mostly incomplete sentences. We would typically not apply stemming and stop-word removal because the title words are generally unique and binary. Bigrams and degree-2 polynomial mappings features are beneficial for product title classification.

Creating labels for such a large dataset would be time-consuming and expensive. As discussed in the class, creating the labels is more effective using Walmart’s Chimera using machine learning, rules and crowdsourcing together. While the crowdsourcing is critical, it should be closely monitored. Crowdsourcing must be coupled with in-house analysts and developers.

Source: <https://blog.acolyer.org/2016/01/28/chimera/>, Class Slides Week 12.

1. [10 pts] Obtaining training data is paramount for a large-scale classification system. You have a limited budget and can’t hire an army of analysts to manually label every single instance. Discuss some strategies for obtaining training data for the classifier.

Solution: Creating labels for such a large dataset would be time-consuming and expensive. As discussed in the class, creating the labels is more effective using Walmart’s Chimera using machine learning, rules and crowdsourcing together. While the crowdsourcing is critical, it should be closely monitored. Crowdsourcing must be coupled with in-house analysts and developers.

The system would iterate as follows:

* Classify the income product items and then use crowdsourcing to continuously evaluate the results and flag cases judged incorrect by the crowd.
* Examine the flagged cases and have fix them by writing new rules, relabeling certain items, and alerting developers.
* Incorporate the newly created rules and the relabeled items into the system, and have the developers fine tune the underlying automatic algorithm.
* If the system refuses to classify certain items due to low confidence, the analysts examine them and create hand-crafted rules as well as training data for those cases. The newly created rules and training data are incorporated into the system and it is run again over the product items.

Source: <https://blog.acolyer.org/2016/01/28/chimera/>, Class Slides Week 12.

1. [5 pts] How would you handle products that are misclassified?

Solution: As mentioned in the previous part of the response, we handle misclassified products by examining the flagged (by crowdsourcing) cases and have fix them by writing new rules, relabeling certain items, and alerting developers. The newly created rules are incorporated and the relabeled items into the system, and the developers fine tune the underlying automatic algorithm. If the system refuses to classify certain items due to low confidence, the analysts examine them and create hand-crafted rules as well as training data for those cases. The newly created rules and training data are incorporated into the system and it is run again over the product items.

Source: <https://blog.acolyer.org/2016/01/28/chimera/>, Class Slides Week 12.

Q5.

1. [10 pts] Sentiment analysis: consider the following review of a restaurant:

*“I took my father out for dinner to Le Bistro on New Year’s Eve. The décor and service were fantastic. We enjoyed the food, especially their French countryside specials and their Chardonnay collections. However, my father thought the menu prices were a bit on the high side. Valet parking was also expensive. Overall, we definitely recommend Le Bistro for special occasions!”*

*Overall rating: 8 stars out of 10*

*“*

Identify the opinion object(s), feature(s), opinion(s), opinion holder(s) and opinion time in this review.

Solution: From the above review, we identify the different components of the review:

* Opinion Object: Le Bistro restaurant
* Opinion Features:
  + Décor and service
  + Food
  + Menu prices
  + Valet parking
* Opinion:
  + The décor and service were fantastic
  + We enjoyed the food, especially the French countryside specials and their Chardonnay collections.
  + Menu prices were a bit on the high side
  + Valet parking was also expensive
  + Recommend Le Bistro for special occasions
* Opinion Holder: The reviewer, the reviewer’s father
* Opinion Time: New Year’s Eve

Source: Class Slides Week 14.

1. [10 pts] Design a sentiment analysis system for restaurant reviews (see example in 5a). Your answer should make use of the techniques discussed in class. The output of the system should assign a sentiment label of Positive or Negative to reviews.

Solution: For the purposes of this question, I’m proposing that I’ll be designing a fresh lexicon-based sentiment analysis system. It will be very similar to the SentiWordNet lexicon-based sentiment analysis system with a slight modification to the weightage assigned (see step 4 below) to the features. Not all the features have equal importance on the review. For example, in the review provided in question 5a, there is 1 opinion object and 4 opinion features. The sentiment on the opinion object will have the highest weightage. When we get to the opinion features, the valet parking and menu prices may be assigned a lesser weightage than the décor and service and the food quality. It is important for the team of data analysts to analyze and assign appropriate weightages to the features. With this understanding, let’s look at the design of the new sentiment analysis system. Consistent with other sentiment analysis systems, we will not remove stop words or apply lemmatization on the reviews.

1. Preprocess, tokenize and POS tag the tokens
2. For each pair of (word, tag), check whether any senti-synsets exist for the same word and the corresponding tag
3. If there is a match, take the first senti-synset and its sentiment score
4. Multiply the senti-synset score to the weightage of the feature (deviation from SentiWordNet method)
5. Aggregate all the scores for the tokens of the review.

Source: Class Slides Week 14.