Course: COSC 4337 Data Science II

Professor: Ricardo Vilalta

TA: Shaila Zaman

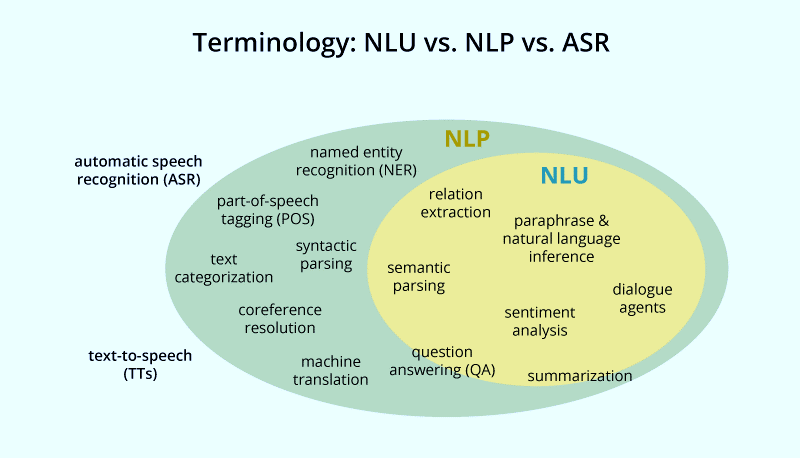
Group member: Kiet Dinh, Giai Tran, Cuong Phan

**The Stanford Natural Language Inference (SNLI) Corpus**

**Data description**

**Source**: The SNLI corpus (version 1.0) is a public dataset from Stanford University Natural Language Processing Group.

**Dataset overview**: the dataset is a collection of 570k human-written English sentence pairs along with their labels for balanced classification task. The SNLI corpus contains threes type of labels: Entailment, Contradiction, and Neutral. It supports the task of understanding, learning, and applying Natural Language Inference (NLI), also known as Recognizing Textual Entailment (RTE).

 So, from the overview of the SNLI corpus dataset, we are trying to address a question about “What is Natural Language Inference (NLI)?”. Basically, from our understanding, we consider NLI is a subset of Natural Language Understanding (NLU) from the outer view. Then, NLU belongs to the root – Natural Language Processing (NLP).

NLI is the task of deciding, given two text fragments, whether the meaning of one text is entailed from another text. The first sentence is Premise while the other one is called Hypothesis.

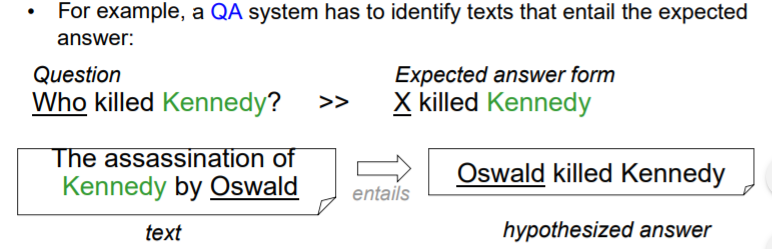
Figure 1. NLP and NLU diagrams

|  |  |
| --- | --- |
| Premise - Sentence 1🡺Hypothesis – Sentence 2 | Entailment |
| Neutral |
| Contradiction |
|  |  |

A screenshot of a cell phone

Description automatically generated

Figure 2. Examples taken from the development portion of corpus (The Stanford NLP Group)

**Column Description**:

* Gold\_label (Categorical): three labels of our classification task – entailment, neutral, contradiction.
* Sentence1\_binary\_parse/Sentence2\_binary\_parse (String): an incomplete binary tree without any POS tag.
* Sentence1\_parse/Sentence2\_parse (String): Part of speech labeled sentences in premise/hypothesis
* Sentence1/Sentence2 (String): the original sentences collected without parsing
* captionID (String): the unique ID of Premise sentence
* pairID (String): the unique ID of Premise – Hypothesis paired sentences
* Label 1-5 (Categorical): they are labeled by the author, group member or volunteers that participate in the project.

**Exploratory Data Analysis & Feature Engineering**

**Import Dataset and Library**

To begin with the preprocessing step, we first import the library Pandas and call its built-in function to read the 3 datasets into corresponding data frames. At first impression, the train dataset contains 550,152 data rows, the development dataset and the testing dataset each contain 10,000 data rows.

**Cleaning the dataset**

The huge collections of sentences and labels are possible to contain invalid values. We perform a function to determine the amount of null values for each column. The sentence\_2 column in the train dataset is notified with 6 rows containing null values. This number is relatively small compared to the entire dataset (0.00001%), so we decide to drop the rows in order to avoid unnecessary issues happening in the later processing.

According to the instruction from the author of the dataset, the class labels are restricted to only 3 values: entailment, contradiction, and neutral. These labels are specifying the meaning relations between sentence 1 (the premise) and sentence 2 (the hypothesis). We first perform a function to check the unique values in the first column (gold\_label) to confirm with the data instruction. The result indicates 4 values including 3 restricted values and a null value. We will be able to apply conditions to each row data and perform a validation check. The operation results in 785 rows with invalid data. Since our goal is to create a classification model on the dataset, missing and erroneous data will affect the model and accuracy. A filtering function is applied to remove rows containing invalid data.

We recognize the two columns captionID and pairID do not provide any helpful information in building our classifier model. Therefore, it is better to drop the two columns to ensure the cleanliness of the data and reduce dimensions. The rest of the columns are converted either to categorical data type they are labels or to string type if they are sentences.

**Visualization**

In the next step, we first count and graph on a bar graph the number of labels between entailment, neutral, and contradiction. Based on the result, we believe that there is a balance between the labels, and it is no longer an issue that needs to be processed and cleaned.

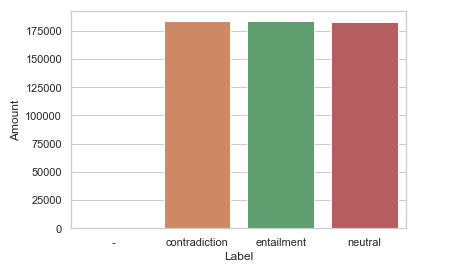
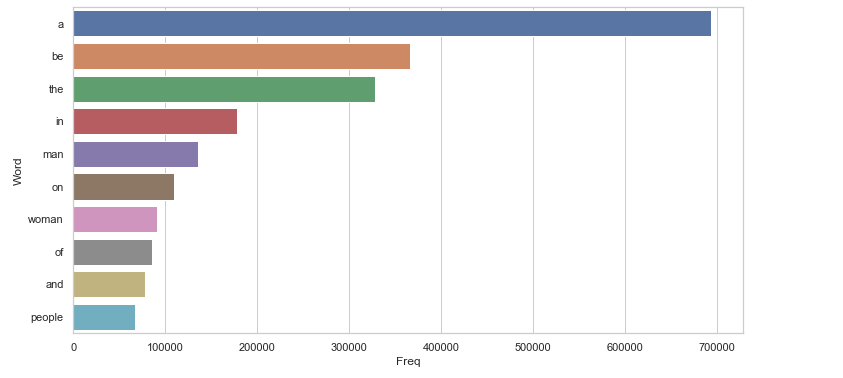


Figure 3. Classification labels distribution.

In each row of data, it is obvious that the sentences contain lowercase and uppercase characters. Because the model may treat a lowercase word differently from the same word containing capital letter and result in incorrect word frequency, it is necessary to reduce the difficulty by avoiding this matter. We perform a quick concatenation to have all sentence\_1 and sentence\_2 columns in the same list and restrict it to unique values only. From the library NLTK, we import the RegexpTokenizerm to parse each sentence and slice them into each word list, traverse through each word, and convert them to lowercase.

Also, to have more understanding on the dataset, we decide to compute the words and word tags frequency and visualize in a bar graph for analysis. Before traversing the entire list to determine the frequency of each word, we perform a function to label the part of speech of the words, and convert each word to its basic form by using the WordNetLemmatizer from nltk.stem, map\_tag, and pos\_tag from nltk. The labeling list will be used to build the model. We import the library Seaborn to visualize the frequency of each word and basic tags in two graphs.

Figure 4. Words with highest frequency.

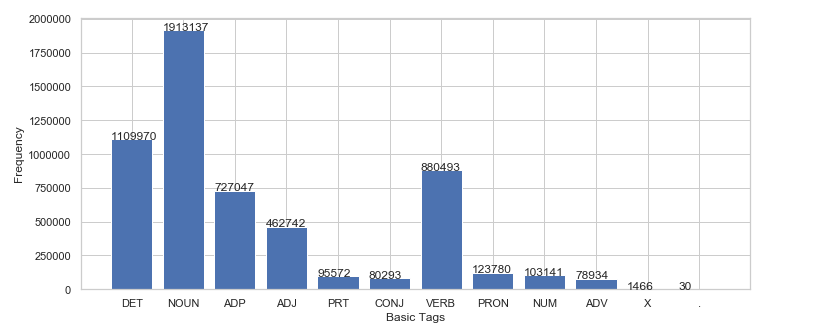


Figure 5. Part-of-speech tags with highest frequency.

The result is surprising, in 628643 unique sentences, the determiner ‘a’ has the highest frequency that is even twice the frequency of the second-place verb ‘be.’ However, in tag graph, the NOUN is the tag that happens to appear most.

**Preprocessing**

**Understanding Part of Speech Tagging (POS Tag):** the most basic models in Natural Language Processing are built based on Bag of Words. However, this kind of model fail to capture the syntactic relations between words.

For example: I like you => “like” is a verb with a positive sentiment.

I am like you => “like” is a preposition with a neutral sentiment.

In order to overcome this disadvantage and improve the efficient of Bag of Words model, we use Part of Speech of each word in a sentence. Then, we build a parse tree and extract relations between words.

**First Approach**: **Embedding with** **Word2Vec**

Natural Language Inference (NLI) focuses mostly on the relationship between hypothesis/premise pairs. To achieve the best results for training models, it is important to transform text data in each pair of sentences into numeric input for algorithms. We apply the same approach in the following diagram to each sentence in every pair for consistency. In this dataset, maintaining the context of the words in sentences is the key to obtain the goal of correct labels classification. For text analysis, we can look at the texts in either as a whole word perspective or in single character perspective. We approach the solution by word embedding technique in which we consider word as a while first to see how well the training models will perform and we can change our technique later. In embedding, the input is words and vector representations are output. Words are represented in a coordinate system where words with the same meanings have similar encoding.

**Word2vec** is a word embedding model that generates a vector space with each unique is mapped into a vector in that space. Words that are located in close proximity to each other have similar context meanings. For example, in the followings two sentences

Sentence 1: The kid is playing soccer with his friends in front of the house

Sentence 2: The child is playing soccer with his friends in front of the house

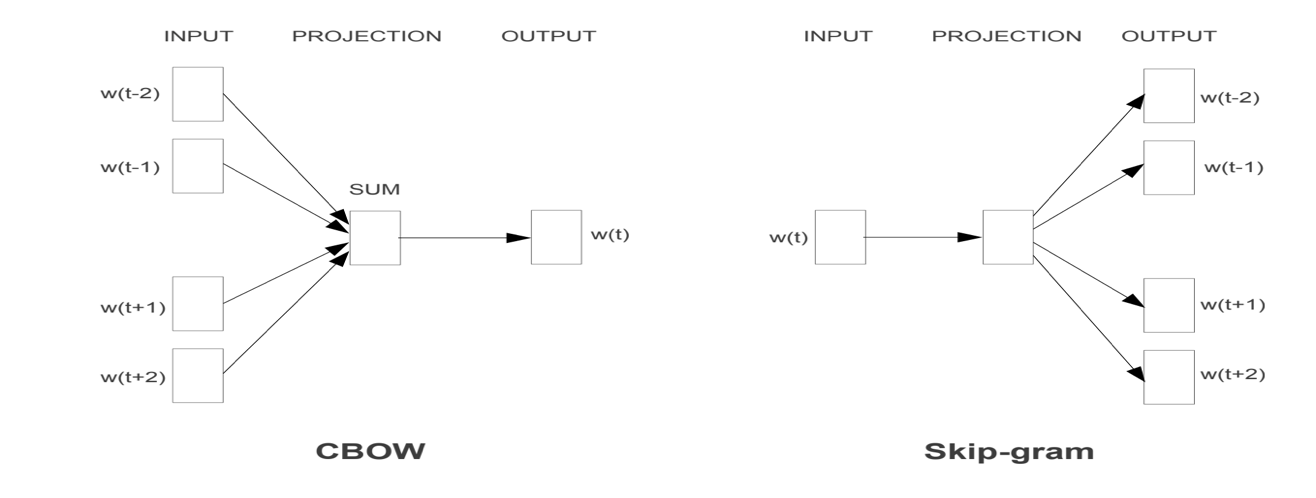
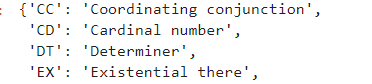
Two words kid and child will have similar embedding. There are two algorithms for generating vectors from the words: Continuous Bag-of-Words Model (CBOW) and Continuous Skip-gram Model. In CBOW, the target word is predicted from the context while in Skip-gram, the context words are predicted from target.

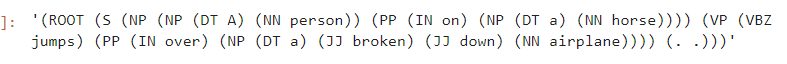
Figure 6: The CBOW & Skip-gram architectures (Mikolov et al. 5).

**Second Approach**: **Probabilistic Methods**

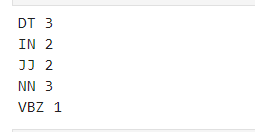
We figure out that the column sentence1 and sentence2 parse are build based on Part-of-Speech Tagging from Penn Treebank Project. The POS Tag system in dataset is more complicated than the implementation in Penn’s project. In addition, we realize there are tags (such as NP, VP, …) which do not exist in Penn’s project after processing the first a hundred samples from column “sentence1\_parse” and “sentence2\_parse”. By analyzing, we decide to use the POS Tag system in Penn’s project since it focuses more on the level of each word instead of the whole phrase which is more complex and harder to implement a model on it for classification task. Penn’s POS Tag system contain 36 unique tags.



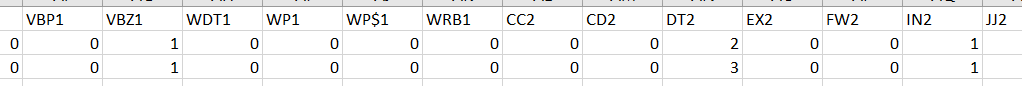
The new data frame for training task is build by calculating the frequency of a tag in a sentence.



From the sentence parse above, we are going to count the frequency of specific tags in the POS Tag system.



Each tag will be a feature in the data frame. Since we have two sentence parse in each sample, tag with a postfix “1” is belong to sentence1 (NN1, DT1, JJ1). Applying the similar concept for sentence2, tags belong to sentence2 will go with postfix “2” (NN2, VBZ2). Our training data frame will end up like this.



Our Preprocessing phrase is finished.

A close up of text on a white background

Description automatically generated

Figure 7. Data Preprocessing Steps Illustration.

**References**

Bar-Haim, Roy, et al. “Semantic Inference at the Lexical-Syntactic Level.” *Semantic Inference at the Lexical-Syntactic Level*, 2007, www.aaai.org/Papers/AAAI/2007/AAAI07-138.pdf.

Mikolov, Tomas, et al. “Efficient Estimation of Word Representations in Vector Space.” *Arxiv.org*, 7 Sept. 2013, arxiv.org/pdf/1301.3781.pdf.

Santorini, Beatrice. “Part-of-Speech Tagging Guidelines for Penn Treebank Project.” *University of Pennsylvania*, 1990, catalog.ldc.upenn.edu/docs/LDC99T42/tagguid1.pdf.

“The Stanford NLP Group.” *The Stanford Natural Language Processing Group*, 2015, nlp.stanford.edu/projects/snli/.