NATURAL LANGUAGE PROCESSING

DIGITAL - 3

Dhruv Garg
16BCE1190

QUESTION 1: WHY WORD EMBEDDINGS ARE BETTER FEATURES AND USED IN DEEP LEARNING?

Why do we need Word Embeddings?

Consider the following sentences: "Have a good day" and "Have a great day". They hardly have different meaning. If we create one-hot vectors for this and try to visualize these encodings, we can think of a 5-dimensional space, where each word occupies one of the dimensions and has nothing to do with the rest (no projection along the other dimensions). This means that "good" and "great" are as different as "day" and "have", which is not true. Thus, our objective is to have words with similar content to occupy close spatial positions. Here comes the idea of generating distributed representations. Intuitively, we introduce some dependence of one word on the other words. The words in content of this word would get a greater share of this dependence.

How word embeddings produce better features?

Traditionally we use bag-of-word to represent a feature. However, they have some limitations such as high dimensional vector, sparse feature. Word embedding is a dense feature in a low dimensional vector. It is proved that word embedding provides a better vector feature.

• Low dimensional : To tackle high dimensional value, word embedding uses a predefined vector space, such as 300, to represent every word. As a pre-defined vector space, the number of dimension (or feature) is fixed no matter how large the corpus is. Comparing to BoW, number of dimension will be increased when the unique words increase.

• Semantic relationship : In general, the word vector encodes semantic relationship among words. It is very important concept since it greatly benefits many NLP problems. Word vectors will be closed if they have similar meaning. For example, buy and purchase will be close.

Continuous Bag-of-Word and Skip-gram

CBOW: uses both n-words before & after target word (w).
skip-gram: uses the opposite approach, which uses the target word to predict 'n' words before & after the target word.

• Negative Sampling : Instead of leveraging all other words as negative label training records, Mikolov proposed to use suitable small amount of training record to train the model, so that the whole operation becomes much faster.

Thus, we can see that word embeddings are the current state-of-art and produce features that are used in Deep Learning for many NLP tasks.

QUESTION 2: WHY IS CO-REFERENCE RESOLUTION IMPORTANT IN NLP TASKS?

Co-reference resolution is the task of grouping all the mentions of entities in a document into equivalence classes so that all mentions in a given class refer to the same discourse entity.

In linguistics, coreference occurs when two or more expressions in a text refers to the same person or thing; they have the same referent, eg. "Bill said he would come".

Co-reference is the main underlying phenomena in the field of syntax: when two expressions are co-referential, the one is usually a full form (antecedent) and the other is an abbreviated form (anaphor).

Many NLP tasks detect attributes, actions, and relations between discourse entities. In order to discover all information about a given entity, textual mentions of that entity must be grouped together. Thus, co-reference resolution is an important step for a lot of higher level NLP tasks that involve natural language ~~processing~~ understanding, such as document summarization, question answering, and information extraction.

"I voted for Nader because he was most aligned with my values", she said.

• Types

ANAPHORA: Use of an expression whose interpretation depends upon another expression in context.

CATAPHORA: Cataphora is the use of an expression or word that co-refers a with a later, more specific, expression in the discourse.

How important is coreference-resolution in Linguistics?

In computational linguistics, co-reference resolution is a well studied problem in discourse. To derive the correct interpretation of a text, or even to estimate the relative importance of various mentioned subjects, pronouns and other referring expressions must be connected to the right individuals.

Algorithms intending to resolve coreferences commonly look first for the nearest preceding individual that is compatible with the referring expression. Algorithms for resolving coreference tend to have an accuracy in the range of 75%. As with many linguistic tasks, there is a tradeoff between precision and recall.

Although co-reference resolution has received much attention, much attention has not been focussed on high-quality features.