Hot Topics in NLP

Daniel Gigliotti

1 Multilingual Semantics

Translation-based approach do not work well since they does not understand: they don't take into account context. We would want instead a way to account for context, understanding the underlying semantics. Once we have the semantics, i.e. the meaning, we can also **scale in multilinguality**: a model that can understand the semantic trained in one language will preserve the same semantic representation when used in other languages.

- **Implicit** understanding of text in latent form:
  - Latent forms/representations are disconnected from explicit, grounded representations of meaning;
  - **Effective**, BUT **hard to interpret**;
  - Tested **extrinsically** in downstream applications;
- **Explicit** understanding:
  - Linked to **lexical-semantic** knowledge resources;
  - **Interpretable** and intelligible;
  - Tested both **intrinsically** (against ground truth semantics) and **extrinsically** (as above).

2 Concepts vs Named Entities

A **concept** is a general idea or notion: it represents a category, class or abstract idea **unambiguously**; A **named entity** is a specific instance of a concept in the real world.

3 How to represent words and senses?

Representing words as vectors yield multiple advantages:

- Easy to use representation;
- Can be visualized;
- Can be combined (to a certain extent);

There are multiple ways we can embed words, but each falls into one of two categories:

- **Explicit** representation;
- **Implicit** representation;

The key differences are:

- **Dimensionality**: Explicit representations, such as one-hot encoding, result in high-dimensional vectors, with a dimension for each unique word in the vocabulary. Implicit representations, on the other hand, create lower-dimensional vectors (e.g., 100-300 dimensions) that capture semantic information.
- **Sparsity**: Explicit representations are typically sparse, as most elements in the vector are zeros except for a single one. Implicit representations are dense, with most elements having non-zero values.
- **Contextual Information**: Implicit representations capture contextual information, meaning that the same word may have different vector representations in different contexts (sense embeddings). Explicit representations, like one-hot encoding, do not capture context.
- **Generalization**: Implicit representations generalize better because they capture semantic and syntactic relationships between words. Explicit representations are limited to counting and presence/absence of words.

Implicit word embeddings have become the standard in many natural language processing tasks because of their ability to capture semantic relationships and their lower dimensionality. They are more efficient and perform better in various NLP applications.

- **Explicit** representations:
  - **One-Hot Encoding**: in this method, each word in the vocabulary is represented as a vector with all zeros except for one element that is set to 1. The position of the 1 in the vector corresponds to the word's index in the vocabulary. This representation is straightforward and explicit, as each word is uniquely identified by its position in the vector space.
  - **Count-Based Methods** (e.g., Document-Term Matrix): count word occurrences in documents or corpora. These methods create a matrix where each row corresponds to a word, and each column corresponds to a document, with the matrix cells containing word counts. These explicit representations capture co-occurrence information between words but may result in high-dimensional and sparse matrices.
  - **TF-IDF** (Term Frequency-Inverse Document Frequency): TF-IDF is another explicit representation that calculates a weight for each word based on its frequency in a document relative to its frequency across all documents. It assigns a numeric value to each word that reflects its importance in a specific document.
- **Implicit** representations:
  - **Word Embeddings**: encode words as continuous-valued vectors in a lower-dimensional space. Word embeddings are learned through neural network-based models such as Word2Vec, GloVe, and fastText. These models capture semantic and syntactic relationships between words by considering their co-occurrence patterns in a large text corpus. The resulting word vectors are dense and capture similarities and differences between words in a distributed representation.

4 PPMI

PPMI stands for "Positive Pointwise Mutual Information" and is a term used in the context of word embeddings and natural language processing. Pointwise Mutual Information (PMI) is a measure that assesses the statistical association between two words in a text corpus. It is often used to build word embeddings or represent the relationships between words in a high-dimensional vector space.

PPMI is a variation of PMI that focuses on the positive associations between words. PMI can take both positive and negative values, but in many natural language processing tasks, negative associations are not as useful as positive associations. PPMI is calculated as follows:

$$PPMI(w_i, w_j) = max(log_2(\frac{P(w_i, w_j)}{P(w_i)P(w_j)}, 0)) \tag{1}$$

Where:

- $PPMI(w_i, w_j)$ is the Positive PMI between words w_i and w_j;
- $P(w_i, w_j)$ is the probability of co-occurrence of words w_i and w_j in a given text corpus;
- $P(w_i)$ and $P(w_j)$ are the individual probabilities of words w_i and w_j occurring in the corpus;

The max function ensures that negative PMI values are truncated to zero, emphasizing only positive associations between words. PPMI is often used in the construction of word co-occurrence matrices or in word embedding models like Latent Semantic Analysis (LSA) to capture the semantic relationships between words in a corpus while downplaying irrelevant or negative associations.