Abstract

Representing texts into vector space is revolutionary to Natural Language Processing, which brings the ability to apply deep learning, the very popular and very powerful machine learning technique, on texts which was previously infeasible. Remarkable works have been done on this topic, namely, *word2vec*, *GloVe*, *doc2vec*, *deepwalk*, and dependency-based word embeddings. Theirs models represent texts into vector space which enabling the computability or compatibility of texts with well-known deep learning models which were previously only applicable for digital data such as images, speeches, etc. *word2vec*/*GloVe* models context distribution of each word via the concept of surround context when a word is used in various text sequences. *dependency-based* word embedding is another work that builds the surround context by traversing through the dependency tree of a sentence, hence, takes care about positionally distant dependencies.

While it is convenient to have word vectors, it is usually not straightforward to compose a document vector from its word vectors. Based on specific tasks, document vectors are learned with certain algorithms or deep learning architecture specialized for the said tasks. *doc2vec* leverages this problem by introducing document-context presence into each word-context and learning the vector representations altogether. However, the implementation does not cover internal structures of the document. Besides, *deepwalk* is another work on context-based vector representation by learning node vectors of a given graph. Similar to *dependency-based* word embedding, *deepwalk* focuses on building the surround contexts of each node by performing random walks through the node.

Document structures can contain relationships including (but not limited to) hierarchy (sections, paragraphs, sentences), discourse (relationships between text-pairs such as agreement, contradiction, or equivalence), and cross-references, though, the previous works only cover a part or none of structural properties of documents.

We aim to build document embedding frameworks that can capture the dependencies within a document in multiple levels of hierarchy: words, sentences, and so on. We develop several methods for capturing those dependencies including context expansion on document hierarchy, *pq*-gram on dependency trees, rhetorical structure, and multi-level contextual features for encoded summarization.

We applied our methods successfully on tasks related to sentence pair modeling and information retrieval.

Keywords: **Deep Learning**, **Representation Learning**, **Information Retrieval**, **Legal Domain**, **Case Law**.