SubGram: Extending Skip-gram Word Representation with Substrings

Abstract

Skip-gram (word2vec) is a recent method for creating vector representations of words ("distributed word representations") using a neural network. The representation gained popularity in various areas of natural language processing, because it seems to capture syntactic and semantic information about words without any explicit supervision in this respect. We propose SubGram, a refinement of the Skip-gram model to consider also the word structure during the training process, achieving large gains on the Skip-gram original test set.

Introduction

Vector representations of words learned using neural networks (NN) have proven helpful in many algorithms for image annotation BIBREF0 or BIBREF1 , language modeling BIBREF2 , BIBREF3 and BIBREF4 or other natural language processing (NLP) tasks BIBREF5 or BIBREF6 .

Traditionally, every input word of an NN is stored in the "one-hot" representation, where the vector has only one element set to one and the rest of the vector are zeros. The size of the vector equals to the size of the vocabulary. The NN is trained to perform some prediction, e.g. to predict surrounding words given a word of interest. Instead of using this prediction capacity in some task, the practice is to extract the output of NN's hidden layer of each word (called distributed representation) and directly use this deterministic mapping INLINEFORM0 of word forms to the vectors of real numbers as the word representation.

The input one-hot representation of words has two weaknesses: the bloat of the size of the vector with more words in vocabulary and the inability to provide any explicit semantic or syntactic information to the

NN.

The learned distributed representation of words relies on much shorter vectors (e.g. vocabularies containing millions words are represented in vectors of a few hundred elements) and semantic or syntactic information is often found to be implicitly present ("embedded") in the vector space. For example, the Euclidean distance between two words in the vector space may be related to semantic or syntactic similarity between them.

Skip-gram Model

The authors of BIBREF7 created a model called Skip-gram, in which linear vector operations allow to find related words with surprisingly good results. For instance INLINEFORM0 gives a value close to INLINEFORM1 .

In this paper, we extend Skip-gram model with the internal word structure and show how it improves the performance on embedding morpho-syntactic information.

The Skip-gram model defined in BIBREF7 is trained to predict context words of the input word. Given a corpus INLINEFORM0 of words INLINEFORM1 and their context words INLINEFORM2 (i.e. individual words INLINEFORM3 appearing close the original word INLINEFORM4), it considers the conditional probabilities INLINEFORM5 . The training finds the parameters INLINEFORM6 of INLINEFORM7 to maximize the corpus probability: DISPLAYFORM0

The Skip-gram model is a classic NN, where activation functions are removed and hierarchical soft-max BIBREF8 is used instead of soft-max normalization. The input representation is one-hot so the activation function is not needed on hidden layer, there is nothing to be summed up. This way, the model is learned

much faster than comparable non-linear NNs and lends itself to linear vector operations possibly useful for finding semantically or syntactically related words.

Related Work

In BIBREF9 was proposed to append part-of-speech (POS) tags to each word and train Skip-gram model on the new vocabulary. This avoided conflating, e.g. nouns and verbs, leading to a better performance, at the cost of (1) the reliance on POS tags and their accurate estimation and (2) the increased sparsity of the data due to the larger vocabulary.

The authors in BIBREF10 used character-level input to train language models using a complex setup of NNs of several types. Their model was able to assign meaning to out-of-vocabulary words based on the closest neighbor. One disadvantage of the model is its need to run the computation on a GPU for a long time.

The authors of BIBREF11 proposed an extension of Skip-gram model which uses character similarity of words to improve performance on syntactic and semantic tasks. They are using a set of similar words as additional features for the NN. Various similarity measures are tested: Levenshtein, longest common substring, morpheme and syllable similarity.

The authors of BIBREF12 added the information about word's root, affixes, syllables, synonyms, antonyms and POS tags to continuous bag-of-words model (CBOW) proposed by BIBREF7 and showed how these types of knowledge lead to better word embeddings. The CBOW model is a simpler model with usually worse performance than Skip-gram.

SubGram

We propose a substring-oriented extension of Skip-gram model which induces vector embeddings from character-level structure of individual words. This approach gives the NN more information about the examined word with no drawbacks in data sparsity or reliance on explicit linguistic annotation.

We append the characters and $ to the word to indicate its beginning and end. In order to generate the vector of substrings, we take all character bigrams, trigrams etc. up to the length of the word. This way, even the word itself is represented as one of the substrings. For the NN, each input word is then represented as a binary vector indicating which substrings appear in the word.

The original Skip-gram model BIBREF7 uses one-hot representation of a word in vocabulary as the input vector. This representation makes training fast because no summation or normalization is needed. The weights INLINEFORM0 of the input word INLINEFORM1 can be directly used as the output of hidden layer INLINEFORM2 (and as the distributed word representation): INLINEFORM3

In our approach, we provide the network with a binary vector representing all substrings of the word. To compute the input of hidden layer we decided to use mean value as it is computationally simpler than sigmoid: DISPLAYFORM0

where INLINEFORM0 is the number of substrings of the word INLINEFORM1 .

Evaluation and Data Sets

We train our NN on words and their contexts extracted from the English wikipedia dump from May 2015. We have cleaned the data by replacing all numbers with 0 and removing special characters except those usually present in the English text like dots, brackets, apostrophes etc. For the final training data we have randomly selected only 2.5M segments (mostly sentences). It consist of 96M running words with the

vocabulary size of 1.09M distinct word forms.

We consider only the 141K most frequent word forms to simplify the training. The remaining word forms fall out of vocabulary (OOV), so the original Skip-gram cannot provide them with any vector representation. Our SubGram relies on known substrings and always provides at least some approximation.

We test our model on the original test set BIBREF7 . The test set consists of 19544 "questions", of which 8869 are called "semantic" and 10675 are called "syntactic" and further divided into 14 types, see Table TABREF4 . Each question contains two pairs of words (INLINEFORM0) and captures relations like "What is to `woman' ( INLINEFORM1 ) as `king' (INLINEFORM2) is to `man' ( INLINEFORM3 )?", together with the expected answer `queen' (INLINEFORM4). The model is evaluated by finding the word whose representation is the nearest (cosine similarity) to the vector INLINEFORM5 . If the nearest neighbor is INLINEFORM6 , we consider the question answered correctly.

In this work, we use Mikolov's test set which is used in many papers. After a closer examination we came to the conclusion, that it does not test what the broad terms "syntactic" and "semantic relations" suggest. "Semantics" is covered by questions of only 3 types: predict a city based on a country or state, currency name from the country and the feminine variant of nouns denoting family relations. The authors of BIBREF13 showed, that many other semantic relationships could be tested, e.g. walk-run, dog-puppy, bark-dog, cook-eat and others.

"Syntactic" questions cover a wider range of relations at the boundary of morphology and syntax. The problem is that all questions of a given type are constructed from just a few dozens of word pairs, comparing pairs with each other. Overall, there are 313 distinct pairs throughout the whole syntactic test set of 10675 questions, which means only around 35 different pairs per question set. Moreover, of the

313 pairs, 286 pairs are regularly formed (e.g. by adding the suffix `ly' to change an adjective into the corresponding adverb). Though it has to be mentioned that original model could not use this kind of information.

We find such a small test set unreliable to answer the question whether the embedding captures semantic and syntactic properties of words.

Rule-Based Baseline Approach

Although the original test set has been used to compare results in several papers, no-one tried to process it with some baseline approach. Therefore, we created a very simple set of rules for comparison on the syntactic part of the test set. The rules cover only the most frequent grammatical phenomenona.

[noitemsep]

adjective-to-adverb: Add ly at the end of the adjective.

opposite: Add un at the beginning of positive form.

comparative: If the adjective ends with y, replace it with ier. If it ends with e, add r. Otherwise add er at the end.

superlative: If the adjective ends with y, replace it with iest. If it ends with e, add st. Otherwise add est at the end.

present-participle: If the verb ends with e, replace it with ing, otherwise add ing at the end.

nationality-adjective: Add n at the end, e.g. Russia INLINEFORM0 Russian.

past-tense: Remove ing and add ed at the end of the verb.

plural: Add s at the end of the word.

plural-verbs: If the word ends with a vowel, add es at the end, else add s.

Our Test Set

We have decided to create more general test set which would consider more than 35 pairs per question set. Since we are interested in morphosyntactic relations, we extended only the questions of the "syntactic" type with exception of nationality adjectives which is already covered completely in original test set.

We constructed the pairs more or less manually, taking inspiration in the Czech side of the CzEng corpus BIBREF14 , where explicit morphological annotation allows to identify various pairs of Czech words (different grades of adjectives, words and their negations, etc.). The word-aligned English words often shared the same properties. Another sources of pairs were acquired from various webpages usually written for learners of English. For example for verb tense, we relied on a freely available list of English verbs and their morphological variations. We have included 100–1000 different pairs for each question set. The questions were constructed from the pairs similarly as by Mikolov: generating all possible pairs of pairs. This leads to millions of questions, so we randomly selected 1000 instances per question set, to keep the test set in the same order of magnitude. Additionally, we decided to extend set of questions on opposites to cover not only opposites of adjectives but also of nouns and verbs.

In order to test our extension of Skip-gram on out-of-vocabulary words, we created an additional subset of our test set with questions where at least one of INLINEFORM0 and INLINEFORM1 is not among the known word forms. Note that the last word INLINEFORM2 must be in vocabulary in order to check if the output vector is correct.

Experiments and Results

We used a Python implementation of word2vec as the basis for our SubGram, which we have made freely available .

We limit the vocabulary, requiring each word form to appear at least 10 times in the corpus and each substring to appear at least 500 times in the corpus. This way, we get the 141K unique words mentioned above and 170K unique substrings (+141K words, as we downsample words separately).

Our word vectors have the size of 100. The size of the context window is 5.

The accuracy is computed as the number of correctly answered questions divided by the total number of questions in the set. Because the Skip-gram cannot answer questions containing OOV words, we also provide results with such questions excluded from the test set (scores in brackets).

Table TABREF18 and Table TABREF19 report the results. The first column shows the rule-based approach. The column "Released Skip-gram" shows results of the model released by Mikolov and was trained on a 100 billion word corpus from Google News and generates 300 dimensional vector representation. The third column shows Skip-gram model trained on our training data, the same data as used for the training of the SubGram. Last column shows the results obtained from our SubGram model.

Comparing Skip-gram and SubGram on the original test set (Table TABREF18), we see that our SubGram outperforms Skip-gram in several morpho-syntactic question sets but over all performs similarly (42.5% vs. 42.3%). On the other hand, it does not capture the tested semantic relations at all, getting a zero score on average.

When comparing models on our test set (Table TABREF19), we see that given the same training set, SubGram significantly outperforms Skip-gram model (22.4% vs. 9.7%). The performance of Skip-gram trained on the much larger dataset is higher (43.5%) and it would be interesting to see the SubGram model, if we could get access to such training data. Note however, that the Rule-based baseline is significantly better on both test sets.

The last column suggests that the performance of our model on OOV words is not very high, but it is still an improvement over flat zero of the Skip-gram model. The performance on OOVs is expected to be lower, since the model has no knowledge of exceptions and can only benefit from regularities in substrings.

Future Work

We are working on a better test set for word embeddings which would include many more relations over a larger vocabulary especially semantics relations. We want to extend the test set with Czech and perhaps other languages, to see what word embeddings can bring to languages morphologically richer than English.

As shown in the results, the rule based approach outperform NN approach on this type of task, therefore we would like to create a hybrid system which could use rules and part-of-speech tags. We will also include morphological tags in the model as proposed in BIBREF9 but without making the data sparse.

Finally, we plan to reimplement SubGram to scale up to larger training data.

Conclusion

We described SubGram, an extension of the Skip-gram model that considers also substrings of input words. The learned embeddings then better capture almost all morpho-syntactic relations tested on test set which we extended from original described in BIBREF7 . This test set is released for the public use.

An useful feature of our model is the ability to generate vector embeddings even for unseen words. This could be exploited by NNs also in different tasks.

Acknowledgment