A new Sentence Completion Set of 6000 questions for natural language models

Jean Argouarc'h
EHESS, Paris & Neurospin, Saclay
jr.argouarch@gmail.com

5/09/2018 version

1 Introduction

Sentence completion is an interesting alternative to word similarity or analogy for benchmarking natural language semantic models. We provide a new set of 6000 sentences , on the same principles as the Microsoft Research Sentence Completion Challenge (MRSCC or MR set, [2], [3],[4]) which has been a very used and useful set since its creation in 2012. The advantages of the MRSCC set are :

- to propose a training corpus of 521 books of the 19th century, in open access, to train the natural language models,
- to define a simple standard test made of one sentence with a missing word and 5 candidate words, giving a 20% chance success. As the five possible sentences are by construction syntactically correct, the test can be used to evaluate semantic models.

We describe in this paper the process used to generate a new set of 6000 sentences and their associated candidate words. The main difference with MR sentences is that to generate them we avoid to use any statistics about the frequency of relations (linear or syntactic context) between words, to prevent from any bias when studying the efficiency of these relations. This issue is explained in [1]. The state-of-the-art accuracy for the MR challenge is about 60%, and about 80% for this new set.

For the clarity of the presentation we will call *question* the basic unit of our set, made of one sentence with a missing word, and five candidate words to take the place of the missing word : the answer word and four impostors.

Our data can be found at http://www.unicog.org/biblio/Category/misc.html.

2 The corpus

The sentences must not come from the MR training corpus used to train the natural language models, but it seems relevant to extract them from similar books, i.e. from books of the same period of the 19th century. In addition, We want to generate sentences with a vocabulary having a good similarity with the words of the MR questions, for the two sets to be comparable. To do so we define and measure a similarity and a coverage ratio between a book and the MR questions. We extract the list of the common word types of the MR questions as a list of tuples w_i (lemma, word-class) and compute the frequency of each tuple in the MR questions, which gives a frequency vector F0 with 3845 integer components.For each book we want to try, we compute the frequency Fb of the same tuples.

The vocabulary similarity is the cosine of the 2 vectors : $cos(F0, Fb) = (F0.Fb)/\|F0\|.\|Fb\|$.

The vocabulary coverage ratio is the ratio of the number of non-zero components of Fb over the number of components of F0.

We have selected 7 books which have a similarity higher than 0.75 and a coverage ratio over 0.70.

Sue, Eugène	The Wandering Jew (n° 3350)
Thackeray, William	The Newcomes: Memoirs of a Most Respectable Family (n° 7467)
Thackeray, William	Vanity Fair (n° 599)
Mühlbach, Luise	Joseph II and His Court: An Historical Novel (n° 3793)
Brontë, Charlotte	Jane Eyre: An Autobiography (n° 1260)
Hawthorne, Nathaniel	Passages from the English Notebooks (n° 7878)
Livingstone, David	Missionary Travels and Researches in South Africa (n° 1039)

3 Question generation

1. Out of the 110 000 sentences of these books, we select about 25000 sentences so that:

   - they contain between 6 and 12 (limits included) common content words,
   - they contain at least one verb,
   - they contain no more than 2 proper nouns.

   At the end we keep only a subset of one sentence every 4 sentences.

2. We process every sentence by randomly choosing a *focus* content word, with a frequency between 10^{-3} and 2.10^{-7} in the MR corpus, which correspond roughly to eliminate the 100 most frequent and the 5 less frequent words. This focus word will be the missing word of the sentence and the answer word of the question.

3. For each sentence we choose randomly, from a vocabulary of 6900 words used in [1], 4 *impostors* i.e. content lemmas of the same class as the focus word (noun, verb, adjective or adverb) with the same frequency constraints as above, and we apply to them the inflection (tense, case, number, etc.) of the focus word. This guarantees that the new sentences will be syntactically correct. Then we generate the 4 new sentences to build the question, and shuffle the set of the 5 candidate sentences. For all candidate words we exclude proper nouns.

4. We keep the first 6000 questions which are stored in the text files :

   - SCS_6000_machine_questions.txt
   - SCS_6000_machine_answers.txt
   - SCS_6000_lm_questions.txt

   These 6000 questions are stored in the same order as for their generation.

5. To provide two statistically equivalent subsets of 3000 questions, we have divided the 6000 questions into 2 subsets : even ranks in subset A, odd ranks in subset B, then by shuffling the ranks in every subset, to get the two lists of numbers from 0 to 5999, which are stored in the text files :

   - SCS_6000_randex_A.txt
   - SCS_6000_randex_B.txt

References

[1] Jean Argouarc'h. Dependency, skip-grams, association tensors and machine learning for sentence completion. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS) Vienna, Austria September 19–21, 2018*, Vienna, Austria, September 2018.

[2] Geoffrey Zweig and Chris JC Burges. The Microsoft Research Sentence Completion Challenge. Technical Report MSR-TR-2011-129, Microsoft Research, 2011.

[3] Geoffrey Zweig and Chris JC Burges. The Microsoft Research Sentence Completion Challenge. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 29–36. Association for Computational Linguistics, 2012.

[4] Geoffrey Zweig, John C. Platt, Christopher Meek, Christopher JC Burges, Ainur Yessenalina, and Qiang Liu. Computational approaches to sentence completion. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 601–610. Association for Computational Linguistics, 2012.