M2 NLP

UE902 EC3: Terminology & Ontology

Project Report

Authors:
Cécile Macaire
Ludivine Robert

November 25, 2020

Contents

1 Introduction

This project takes its place in the terminology that studies terms and their use. The goal is to develop a term identification system for a specific domain. At our level we will be focused on monolingual corpus.

In the field of Natural Language Processing (NLP), a terminology is a coherent set of terms that constitutes the vocabulary of a domain. The terms are, in this case, lexical items (single or complex) used in discourse and are subjective to linguistic variations.

Because there exist numerous domains in the world, working on terminology can help to understand and communicate better in a specific discipline. It may also be an advantage to use terminology in certain computer or information tasks.

Our team is composed of Cécile which have more background in computer sciences and Ludivine which have more background in linguistics. As the choice of the domain was free, we decided to work on a branch of NLP which is Text-to-Speech (TTS) synthesis. We selected this area due to the fact that our software project (part of the master) is about a Multilingual Text-to-Speech system.

2 Methodology

2.1 Data collection

The first step was to build a corpus to train our model. After selecting twenty two articles from our chosen domain, in *.pdf* format, we converted them in *.txt* format and cleaning them manually (to be used later). The cleaning process involved removing the mathematical formulas, tables and references. The text files will be our train corpus. We also took the abstract from two other articles for our test corpus.

2.2 Lexicon creation

The second step was to set up our lexicon with the extracted terms from our corpus. In order to facilitate the task, we did the extraction part automatically using `TermSuite`[1][1]. `TermSuite` is a toolbox for terminology extraction and multilingual term alignment, developed at the University of Nantes by the research lab LS2N, UMR 6004 CNRS.

In particular, `TermSuite` allows to specify particular features that will be useful for this project. We have chosen to save the pattern, lemma and frequency associated with each term. Moreover, a human manual filtering was needed to get a list of good quality terms. Figure 1 shows the beginning of our final lexicon.

#	type	pattern	pilot	lemma	freq
	T	N N	1D Convolution	1d convolution	
	T	N N	1D convolutional	1d convolutional	
9	T	A N	acoustic model	acoustic model	72
497	T	A N N	acoustic model training	acoustic model training	5
2346	T	A	acoustic-phonotactic	acoustic-phonotactic	1
478	T	N N	Adam optimization	adam optimization	5
225	T	N N	Adam optimizer	adam optimizer	9
676	T	A A N	adaptive cepstral analysis	adaptive cepstral analysis	4
279	T	N N	adversarial loss	adversarial loss	8
334	T	N N N	adversarial speaker classifier	adversarial speaker classifier	7
	T	A N N	artificial human speech	artificial human speech	
1345	T	N	ASR	asr	
438	T	N N	ASR performance	asr performance	5
82	T	N N	attention alignments	attention alignment	19
121	T	N N	attention block	attention block	14
	T	N N N	attention context vector	attention context vector	

Figure 1: Beginning of the lexicon file data.

2.3 Identification systems

The third step, the most important, was to developed identification systems relying on our lexicon. We implemented them using Python and associated library such as Spacy[2] and Pandas[3].

[1] `http://termsuite.github.io/documentation/terminology-tsv-output//`
[2] `https://spacy.io/`
[3] `https://pandas.pydata.org/`

2.3.1 Rule-based approach

We based our system on dictionary lookup and heuristics.

The main idea of our approach is to identify the terms by looking into the list of terms created before and to the rules we constructed.

Our model annotates the text in several steps:

1. Extraction of the terms from the lexicon and lemmatization with Spacy,
2. Extraction of each element from the text and lemmatization with Spacy,
3. The previously defined rules are applied to the text and elements identified as terms are annotated between square brackets,
4. Finally, if a term from the lexicon is visible in the text, it is also annotated between square brackets.

Lemmatizing terms and text makes it possible to homogenise elements (e.g. *Model* and *model* have the same lemma) and to manage plural forms (e.g. *vector* and *vectors* have the same lemma).

The rules we have defined are explained below.

Our corpus has the particularity of possessing terms of the form <Noun-Prep-Noun> (e.g. *text-to-speech, sequence-to-sequence, phoneme-by-phoneme*, etc.). These terms can be followed by 1 or 2 nouns, or 1 adjective and 1 noun. The rule will therefore annotate these specific patterns (e.g. *text-to-speech synthesis system, end-to-end pipeline, end-to-end neural speech*, etc.).

Our corpus also possesses terms with 2 words separated by a dash (e.g. *encoder-decoder, HMM-based*, etc.). If these terms are followed by these patterns — <Noun, Noun, Noun>, <Noun, Noun>, <Verb, Noun>, <Adjective, Noun>, <Noun> — we consider also them as terms (e.g. *HMM-based speech synthesis system, n-gram language modeling, grapheme-based model*, etc.).

Some words are not considered as terms unless they follow a specific pattern. Here we are talking about the words *data, voice, datum, speaker, dataset, database, feature and corpus* which are considered as terms when they are preceded by a noun or adjective, and/or followed by a noun (e.g. *feature map, audio data, character database*, etc.).

Finally, if a term is preceded by a specific adjective, and/or is followed by a specific word, the whole is considered as one term. Some of the adjectives are *autoregressive, multilingual, supervised*, etc. and some of the words are *activation, adaptation, network, transcription*, etc.. Indeed, these words, alone, are not considered as terms. For example, *phoneme representations* and *autoregressive models* will be terms.

Figure 2 shows an extract from an annotated text after applying the rules.

```
We conduct experiments on the [LJSpeech dataset] to test [FastSpeech].
The results show that in terms of [speech quality], [FastSpeech] nearly matches
the [autoregressive Transformer model].
Furthermore, [FastSpeech] achieves 270x speedup on [mel-spectrogram generation]
and 38x speedup on final [speech synthesis] compared with the [autoregressive
Transformer TTS model], almost eliminates the problem of word skipping and
repeating, and can adjust [voice speed] smoothly.
We attach some [audio files] generated by our method in the supplementary
materials.
```

Figure 2: Extract of an annotated file.

2.3.2 Sequence taggers

We trained an IOB tagger and applied a Part-Of-Speech (POS) tagger (thanks to Spacy) on the corpus annotated by our rule-based approach.

The IOB tagger consists of 3 tags:

- B: Beginning of multiword term.
- I: Inside a multiword term (not at the beginning).
- O: outside a term.

In our methodology, whenever an element of the annotated text contains an open bracket, it will be tagged B and the rest of the words will be tagged with I until the script finds the closed bracket. The rest of the text is tagged with an O.

Figure 3 is the annotated text from Figure 2 tagged into IOB form.

```
We (O) conduct (O) experiments (O) on (O) the (O) [LJSpeech (B) dataset (I)]
to (O) test (O) [FastSpeech (B)].
The (O) results (O) show (O) that (O) in (O) terms (O) of (O) [speech (B)
quality (I)], [FastSpeech (B)] nearly (O) matches (O) the (O) [autoregressive
(B) Transformer (I) model (I)].
Furthermore, (O) [FastSpeech (B)] achieves (O) 270x (O) speedup (O) on (O)
[mel-spectrogram (B) generation (I)] and (O) 38x (O) speedup (O) on (O) final
(O) [speech (B) synthesis (I)] compared (O) with (O) the (O) [autoregressive
(B) Transformer (I) TTS (I) model (I)], almost (O) eliminates (O) the (O)
problem (O) of (O) word (O) skipping (O) and (O) repeating, (O) and (O) can
(O) adjust (O) voice (O) speed (O) smoothly. (O)
We (O) attach (O) some (O) [audio (B) files (I)] generated (O) by (O) our (O)
method (O) in (O) the (O) supplementary (O) materials. (O)
```

Figure 3: Extract of an annotated and IOB tagged file.

We also made the choice to tag the annotated and IOB tagged text with POS tags. Figure 4 bellow shows the final tagged text.

```
 We PRON (O) conduct NOUN (O) experiments VERB (O) on ADP (O) the DET (O)
[LJSpeech NOUN (B) dataset NOUN (I)] to PART (O) test NOUN (O) [FastSpeech
PROPN (B)] .
The DET (O) results VERB (O) show NOUN (O) that SCONJ (O) in ADP (O) terms
NOUN (O) of ADP (O) [speech NOUN (B) quality NOUN (I)] , [FastSpeech PROPN
(B)] nearly ADV (O) matches NOUN (O) the DET (O) [autoregressive ADJ (B)
Transformer VERB (I) model NOUN (I)] .
Furthermore ADV , (O) [FastSpeech PROPN (B)] achieves VERB (O) 270x NUM (O)
speedup NOUN (O) on ADP (O) [mel PROPN - spectrogram PROPN (B) generation
NOUN (I)] and CCONJ (O) 38x NOUN (O) speedup NOUN (O) on ADP (O) final ADJ
(O) [speech NOUN (B) synthesis NOUN (I)] compared VERB (O) with ADP (O) the
DET (O) [autoregressive ADJ (B) Transformer VERB (I) TTS PROPN (I) model NOUN
(I)] , almost ADV (O) eliminates NOUN (O) the DET (O) problem NOUN (O) of ADP
(O) word NOUN (O) skipping NOUN (O) and CCONJ (O) repeating VERB , (O) and
CCONJ (O) can VERB (O) adjust NOUN (O) voice NOUN (O) speed NOUN (O) smoothly
ADV . (O)
We PRON (O) attach NOUN (O) some DET (O) [audio NOUN (B) files NOUN (I)]
generated VERB (O) by ADP (O) our DET (O) method NOUN (O) in ADP (O) the DET
(O) supplementary ADJ (O) materials NOUN . (O)
```

Figure 4: Extract of an annotated and IOB + POS tagged file.

3 Evaluation

3.1 Metrics

To evaluate our annotations, we used two abstracts from new scientific articles of our domain as test set. The evaluation has been performed using f-measure.

We compared the output of our model with manually annotated test texts (expected output) to see if our system over-annotated or missed some annotations.

Common indexes have been used:

- Precision: % of selected items that are correct.
- Recall: % of correct items that are selected.

$$precision = \frac{pertinent\ annotations}{all\ annotations\ occured}$$

$$recall = \frac{pertinent\ annotations}{all\ pertinent\ annotations}$$

We combined both to calculate the f-measure, which corresponds to the harmonic mean of precision and recall:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

To calculate the f-measure, we focused on the terms annotated in the expected output and in our model's output.

A term which is annotated in the expected output and in our tool's output is considered as true positive. A term which is annotated in the expected output but not in our tool's output is considered as false negative. A term annotated in our tool's output but not in the expected output is considered as false positive. Finally, a term which is not annotated in our tool's output and not in the expected output is considered as true negative. True negatives are not taken into account by f-measure, rightfully within the scope of our project.

3.2 Results

	Automatic annotation		
		+	-
Manual annotation	+	13+18	0+1
	-	4+2	-

Table 1: Confusion matrix of the evaluation results.

$$precision = \frac{31}{38}$$

$$recall = \frac{31}{32}$$

$$F_1 = 2 * \frac{(31/38) * (31/32)}{(31/38) + (31/32)} \simeq 0.88$$

As shown in Table 1 and after doing the calculations, the average f-measure is almost 90%, which is a good result.

From our text set, the main error we faced was the non-recognition of some newer proper noun such as *Wave-Tacotron* or *Human Language Technology*, as shown in Figure 7. This means that the main weakness of the program is the omission of certain annotations of terms, due to the exceptions that occur in the terms of this specific domain.

meaning or resolve into the [target language]. To achieve such, there is a need to adopt coordinated cooperation of the separate [Human Language Technology] components. In particular, the most significant elements in a [speech translation system] encompass the [automatic speech recognition], [machine translation] as well as [text to speech]. With this understanding, the current paper explores

Figure 5: Expected output

meaning or resolve into the [target language]. To achieve such, there is a need to adopt coordinated cooperation of the separate Human Language Technology components. In particular, the most significant elements in a [speech translation system] encompass the [automatic speech recognition], [machine translation] as well as [text to speech]. With this understanding, the current paper explores

Figure 6: Tool's output

Figure 7: Sample of the evaluation outputs

4 Conclusion

This project has been a good experience to understand and manipulate terminology.

Several tasks in the implementation of this project required more attention and were more or less time-consuming:

- the construction of the corpus, so that it is perfectly usable (deletion of certain elements, certain irrelevant parts, etc.),
- the exceptions which occur in this specific field, such as expressions containing a dash,
- the identification of the rules specific to our corpus,
- the manual checking of each annotated text to correct exceptions.

The results of the evaluation are quite satisfactory but some improvements to the system would make it totally reliable, such as working on a larger corpus, which would provide a more detailed lexicon.

References

[1] Damien Cram and Béatrice Daille. "Terminology extraction with term variant detection". In: *Proceedings of ACL-2016 System Demonstrations*. 2016, pp. 13–18.