Know-Center at SemEval-2017 Task 10: Sequence Classification with the CODE Annotator

Roman Kern
Know-Center GmbH
Inffeldgasse 13
Graz, 8010, Austria
rkern@know-center.at

Stefan Falk
Know-Center GmbH
Inffeldgasse 13
Graz, 8010, Austria
sfalk@know-center.at

Andi Rexha
Know-Center GmbH
Inffeldgasse 13
Graz, 8010, Austria
arexha@know-center.at

Abstract

This paper describes our participation in SemEval-2017 Task 10, named *ScienceIE (Machine Reading for Scientist)*. We competed in Subtask 1 and 2 which consist respectively in identifying all the key phrases in scientific publications and label them with one of the three categories: Task, Process, and Material. These scientific publications are selected from Computer Science, Material Sciences, and Physics domains. We followed a supervised approach for both subtasks by using a sequential classifier (CRF - Conditional Random Fields). For generating our solution we used a web-based application implemented in the EU-funded research project, named CODE. Our system achieved an F1 score of 0.39 for the Subtask 1 and 0.28 for the Subtask 2.

1 Introduction

Information Retrieval (IR) systems for scientific publications face different challenges compared to the standard approaches. This, mainly is due to the unavailability of the whole text from reviewed papers and the vague specification of the searching information. The identification and the extraction of the key phrases from such articles can partially overcome the limits described above by allowing search engines to access and use them as text features. Furthermore, the classification of the key phrases as a Task, a Process, or a Material, can help the researchers to correctly specify the type of information they are seeking.

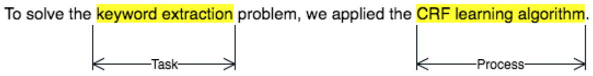

To solve the keyword extraction problem, we applied the CRF learning algorithm.

Figure 1: Example of a keyphrase with its associated label

The Subtasks 1 and 2 of Task 10 (Augenstein et al., 2017) in SemEval-2017 named *ScienceIE (Machine Reading for Scientist)*, tackle the aforementioned problems. This task consists in identifying (Subtask 1) and labeling (Subtask 2) all the key phrases in scientific publications from Computer Science, Material Science, and Physics.

For training and evaluating this task, it was provided a set of scientific papers together with the annotated key phrases and their associated labels. The annotations were represented with their start and end offsets in the text. The labels associated with each annotation can be from one of the three options: Task, Process, and Material. The example in Figure 1 illustrates the given dataset.

We followed a supervised approach for both subtasks. More specifically, we trained a sequential classifier CRF - Conditional Random Fields (Lafferty et al., 2001) and fed it with grammatical and text features. The model built from this classifier represents our solution for identifying and labeling the key phrases.

The rest of the paper is organized as follows. In the next section we describe our system's details. In section 3 we show the results of our systems and compare it with the other participants in the challenge. We end with section 4 summing up the con-

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 961–964,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

clusions and foreseeing our future work.

2 System Description

In the ScienceIE (Machine Reading for Scientist) we have followed a supervised approach. For classifying a certain number of elements as key phrases and label them, we use a CRF (Conditional Random Field) classifier. Our system is part of an open-source tool[1] that has been developed within a EU funded research project, named CODE[2]. This system is a web-based application, which allows to quickly annotate textual corpora imported directly from Mendeley[3], an E-Mail server or a Zip-file containing Brat annotations.

Once the corpus has been imported, it is automatically pre-processed and indexed using a semantic search engine. In order to make use of an automatic annotation of a corpus, a model needs to be trained. This is conducted using solely the web interface of the tools, see Figure 2 for a screenshot of the configuration panel where the model can be trained. The submitted runs have been generated using exclusively the CODE Annotator tool, only a slight modification of the Brat annotation files as supplied by the organisers were necessary.

2.1 Pre-Processing

Given the individual tokens and sentences we apply a light pre-processing on the text. At first we apply a part-of-speech tagger, namely OpenNLP[4], to derive the word form of each word within the sentence. As our pre-processing pipeline is designed to work with multiple languages, with each having its own dedicated tagset, we defined our own uniform POS tagset. This tagset consists of just 14 different word forms, e.g. proper nouns and common nouns (including the tags that indicate plural) are all unified into a single noun tag. We store the original POS tags together with the unified tags within an internal representation of the text.

2.2 Feature Generators

We used a series of feature generators that operate on the pre-processed sentences to create features, which are then fed to the classifiers.

Tokens The token feature generator directly encodes the individual words as features, following a bags of words approach. This generator offers the configuration parameter to optionally normalise the tokens, i.e. to bring them into a lower-case representation. For the submitted runs, we used the raw tokens without further normalisation.

Token Character The first and last characters of a word are often indicative of its semantic and grammatical function. Therefore we crafted a feature generated that generates character n-grams from the prefix and suffix of the tokens. This generator provides options on the length of the generated n-gram features. We finally ended up using 1, 2 and 3-gram features, which are additionally normalised by bringing them into a lower case representation.

Token Shape In many different domains entities are often abbreviations or specific words using combinations of special characters and numbers. To capture this, we designed a feature generator that maps a word into a representation that should reflect the words shape. All characters of a word are mapped to a sequence of characters that represent: upper case, lower case, special characters, and numbers. For each word we create two representations: i) adjacent mappings are conflated into a single character, ii) adjacent mappings are merged into a single character, but additionally the number of merged characters is appended. For example, the entity "`NFAT/AP-1`" will yield two features: "`A/A-x`" and "`A4/1A2-1x1`".

Token POS Tag This feature generator simply adds the POS tagging to the set of features. We used our unified tag set instead of the Penn Treebank tagset, which is used by the POS tagger.

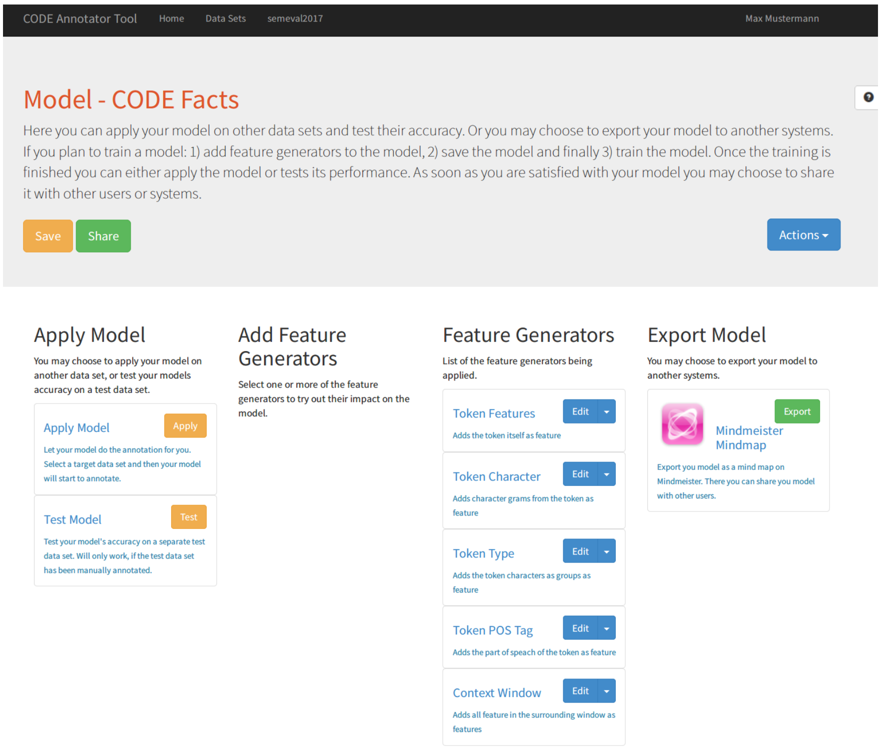
Context Window This feature generator takes the features from the surrounding words and adds them to the feature set of the word in focus. We one can specify the size of the sliding window - with the left and right window size individually. Based on pre-

[1] http://code-annotator.know-center.tugraz.at/

[2] http://code-research.eu/

[3] http://mendeley.com/

[4] https://opennlp.apache.org/



Model - CODE Facts

Here you can apply your model on other data sets and test their accuracy. Or you may choose to export your model to another systems. If you plan to train a model: 1) add feature generators to the model, 2) save the model and finally 3) train the model. Once the training is finished you can either apply the model or tests its performance. As soon as you are satisfied with your model you may choose to share it with other users or systems.

Apply Model

You may choose to apply your model on another data set, or test your models accuracy on a test data set.

Apply Model

Let your model do the annotation for you. Select a target data set and then your model will start to annotate.

Test Model

Test your model's accuracy on a separate test data set. Will only work, if the test data set has been manually annotated.

Add Feature Generators

Select one or more of the feature generators to try out their impact on the model.

Feature Generators

List of the feature generators being applied.

Token Features

Adds the token itself as feature

Token Character

Adds character grams from the token as feature

Token Type

Adds the token characters as groups as feature

Token POS Tag

Adds the part of speach of the token as feature

Context Window

Adds all feature in the surrounding window as features

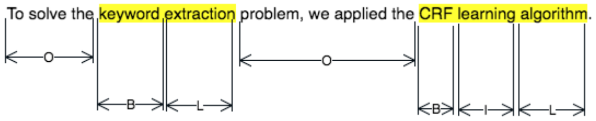
Export Model

You may choose to export your model to another systems.

Mindmeister Mindmap

Export you model as a mind map on Mindmeister. There you can share you model with other users.

Figure 2: Screenshot of the CODE annotator tool, where users can tweak the model, trigger the learning process, evaluate the model on test corpora or apply a model on a corpus.



To solve the keyword extraction problem, we applied the CRF learning algorithm.

Figure 3: An example of the application of a sequence of words labelled with a BILOU encoding.

liminary test we opted to use a very short window of just 1 word to the left and right of each word.

2.3 Classification

To label words as part of key phrases, we followed a sequence classification approach. Here a single sentence is seen as a sequence of items, which are all assigned to a label.

Key Phrases As key phrases may consists of multiple words and multiple key phrases may directly succeed each other, one needs a labelling scheme that cater for this cases. The most common encoding schemes are "`BIO`" and "`BILOU`". Based on preliminary tests we opted for the latter, which should be more expressive, but may yield worse results in some scenarios.

The "`BILOU`" encoding scheme refers to classify each of the token as either: B) beginning of a (multi-token) key phrase, I) used for all tokens inside a (multi-token) key phrase, L) for the last token of a (multi-token) key phrase, O) used for tokens outside of a key phrase (i.e. all tokens not being part of a key phrase), and finally U) used for key phrases consisting of just a single token. See Figure 3 for an example how the labels are constructed.

To fit the current classifier with the Subtask 2, we encode each word contained in a key phrase as a concatenation of the key phrase's label with the corresponding "`BILOU`" encoding. Consider the key phrase "`keyword extraction`" in the example 3 and let's assume that its label is "`Task`". We would encode the word "`keyword`" as "`Task-B`" and the word "`extraction`" as "`Task-L`". The encoding returned from the CRF algorithm would determine then the label of the key phrase.

Key Phrase Classification Algorithm We used the Conditional Random Field (CRF) algorithm as supplied by the Mallet[5] library. Mallet does allow to specify the order of the random field. Due to the small size of the training data set we were able to use a fully-connected model. Furthermore we were able to train the model until convergence, without the need to stop at a predefined threshold.

3 Results

Here we describe the results of the challenge and compare all the other participating teams. Table 1 shows the results for the two subtasks we have participated in.

System	**F1 score for Subtask 1**	**F1 score for Subtask 2**
TIAL_UW	0.56	0.44
s2_end2end	0.55	0.44
PKU_ICL	0.51	0.38
TTI_COIN	0.50	0.39
NTNTU-1	0.47	0.34
WING-NUS	0.46	0.33
SciX	0.42	0.21
IHS-RD-BELARUS	0.41	0.19
Know-Center	**0.39**	**0.28**
LIPN	0.38	0.21
SZTE-NLP	0.35	0.28
LABDA	0.33	0.23
NTNU	0.30	0.24
NITK_IT_PG	0.30	0.15
HCC-NLP	0.24	0.16
Surukam	0.24	0.1
GMBUAP	0.08	0.04

Table 1: Official results for the Subtask 1 and 2 of the Task 10 in Semeval-2017, named *ScienceIE (Machine Reading for Scientist)*

As illustrated, we have achieved a F1 score of 0.39 for the Subtask 1 and 0.28 for Subtask 2. The best performing team managed to achieve an F1 score of 0.56 and 0.44 respectively for each subtask. We ranked in the 9-th place for the Subtask 1 and 7-th for the Subtask 2.

4 Conclusions and Future Work

In this paper we presented our system for the SemEval-2017 Task 10, named *ScienceIE (Machine Reading for Scientist)*. We competed in Subtask 1 and 2, which consist, respectively, in identifying all the key phrases in scientific publications and label them. We achieved an F1 score of 0.39 for the Subtask 1 and 0.28 for the Subtask 2.

Our plan for the future work is to extend the set of used features and analyse their impact. Furthermore we intend to consider different classifications algorithm and tune their parameters.

Acknowledgments

The Know-Center GmbH Graz is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

[Augenstein et al.2017] Isabelle Augenstein, Mrinal Kanti Das, Sebastian Riedel, Lakshmi Nair Vikraman, and Andrew McCallum. 2017. SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications. In *Proceedings of the International Workshop on Semantic Evaluation*, Vancouver, Canada, August. Association for Computational Linguistics.

[Lafferty et al.2001] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[5] `http://mallet.cs.umass.edu/` (Version 2.0.7)