# Dialogue and Narrative: SemEval 2017 Task 10 - Extracting Key Phrases from Scientific Publications

**Stefan Radic Webster**
University of Bristol
hn19405@bristol.ac.uk

## Abstract

The paper describes an approach for SemEval's 2017 Task 10 to extract key phrases from scientific publications (subtask A), and then classify the key phrases (subtask B). The focus was to produce end-to-end systems using evaluation scenario 1 where no annotation information is available during the test phase. First, a Hidden Markov Model was developed which yielded a micro F1 score of 0.44 and 0.31 for subtasks A and B, respectively. Furthermore, a Bi-LSTM-CRF model was trained using Flair contextual string embeddings which scored higher than the entries to the shared task with an F1 score of 0.62 and 0.46 for subtask A and B, respectively. The code is available from GitHub [1].

## 1 Introduction

The shared task of machine reading for scientists [2] at SemEval 2017 involved the automatic extraction of key phrases from scientific publications, classifications of the types of key phrases and identification of relations between key phrases. Key phrases capture a documents main topics and ideas as a summarisation exercise which can be used in a range of downstream tasks (Kim et al., 2010). Scientific research involves addressing a specific task, studying a process using various materials and as such key phrases can be classified into Task, Process or Material. Process key phrases relate to scientific models and algorithms, key phrases should be labelled task if they describe the application, end goal or problem and material key phrases identify the resources used. Automatic extraction of key phrases and their labels can be used as metadata for search engines to assist readers to search vast databases of papers (Augenstein et al., 2018).

Key phrases extraction and classification is related to the NLP task of named entity recognition but poses a more challenging task as key phrases can vary between domains, lack clear contextual features and can consist of many tokens. As a result, the training of an end-to-end system was preferential over traditional methods of hand-crafting features to select potential key phrases to be fed into machine learning models (Kim et al., 2013).

## 2 Task description and data

Subtask A involves key word identification whilst subtask B classifies these key phrases into Task, Process or Material. Evaluation scenario 1 (Augenstein et al., 2018) was followed for the task where the test sentences have no annotation during subtask B evaluation. Subtask B therefore consisted of the identification and then subsequent classification of key phrases and as a result, the performance of subtask B can never exceed subtask A.

The corpus for the task consisted of 500 paragraphs from journal articles from ScienceDirect publications. The corpus was divided into 350 for model training, 50 development documents for model optimisation with the remaining 100 for testing. The paragraphs from the publications were provided in plain text format along with annotations files. The double-annotated documents were first annotated by student annotators and then checked by an exert-annotator (Augenstein et al., 2018).

Evaluation was conducted exactly against the gold standard annotations with the metrics precision, recall and F1-score being computed and the micro-average (weighted) F1 score used for comparison. F1 score is an evaluation metric primarily

[1] https://github.com/sradicwebster/dialogue_and_narrative/

[2] https://scienceie.github.io/index.html

used for classification tasks and is preferred over accuracy for NLP tasks as it takes class distribution into consideration.

### 2.1 Data Preparation

Each document in the corpus (consisting of training, development and testing documents) had a text file with the paragraph text and an annotation (.ann) file which identified the span of the key phrases along with the key phrase types.

The text files were tokenised using NLTK TreebankWordTokenizer which identifies the span of the identified tokens. This allowed the span annotations to be aligned with the tokenised text and the tokens were labelled being at the beginning, inside or outside a key phrase as per the 'BIO' encoding convention. Using 'BIO' encoding converts the task into a raw labelling problem of assigning a single label to each word in a sequence.

### 2.2 Baselines

Simple models to find the upper bound and lower baselines were described by (Augenstein et al., 2018). The upper bound was found by converting the annotation files into tokens and back to span annotations. An F1 of 0.85 was found as the upper bound for key phrase extraction and classification showing the information loss due to splitting the paragraphs into sentences and tokenisation. The lower baseline was found by assigning a random label to each token which gave an F1 score of 0.03 for key phrase identification, highlighting the difficulty of the task. In addition, key phrase classification depends on the correct key phrase identification which lowers the random baseline F1 score to 0.01.

For both key phrase identification and classification, the most common tag is 'O' meaning the token is not part of a key phrase. Classifying all the tokens as 'O' gave an F1 score of 0.28 and 0.21 for key phrase identification and classification, respectively, which was used as the baseline for the task.

| Baseline | Subtask A | Subtask B |
|---|---|---|
| LB random | 0.03 | 0.01 |
| LB all outside | 0.28 | 0.21 |
| UB | 0.85 | 0.85 |

Table 1: Upper bound and lower baselines.

## 3 Modelling and Results

### 3.1 NLTK Hidden Markov Model

NLTK was used to train a Hidden Markov Model (HMM) which uses local sequence information. HMMs have two probability distributions a transition and emission distribution. Transition distributions model the probability of a tag given the previous tag (for example how likely a word being the beginning of a key phrase will follow a word not part of a key phrase) and emission distributions which is probability of an observed word given a tag (for example how likely a word is given that we know this word should be part of a key phrase) (Jurafsky and Martin, 2000).

The HMM were trained using the supplied training data and evaluated using the test sentences. For key phrase identification by tagging the tokens to be labelled being outside, at the beginning or inside a key phrase, an F1 score of 0.443 was found. Subsequently, a HMM was trained to identify and classify the key phrases as process, material or tasks which had an F1 score of 0.314.

The HMM for identifying and classifying key phrases from scientific documents performed relatively well compared to entries to the SemEval task in 2017, and would have placed in the top half of the leaderboard. Entrants who produced superior models used neural network architectures (RNN, LSTM  CNN) and conditional random fields which have been shown to produce superior results for sequence labelling tasks (Peters et al., 2017; Prasad and Kan, 2017).

### 3.2 Flair Deep Learning

The NLP framework Flair [3] was used for training end-to-end neural network models for the tasks. The training, development and testing files (labelled tokens after preprocessing) were input into the Flair ColumnCorpus to create a corpus for training and evaluating the models. The number of sentences in the training, development and testing data were 1386, 367 and 748, respectively.

The Flair text embedding library contains state-of-the-art word embeddings, such as GloVe, BERT and ELMo and also facilitates combining different word embeddings using StackedEmbeddings. The combination of embeddings chosen which gave best results was using GloVe and Flair

[3] https://github.com/flairNLP/flair

| Hyperparameter | Value |
|---|---|
| Hidden states | 256 |
| Drop out | 0.1 |
| Learning rate | 0.17 |
| Mini-batch size | 16 |

Table 2: Flair Bi-LSTM-CRF hyperparameters.

Embeddings (both forward and backwards models).

The task of key phrase extraction is a sequence labelling task. The contextual embeddings were passed to a Bidirectional Long Short Term Memory with Conditional Random Field architecture (Bi-LSTM-CRF) in Flair's SequenceTagger to solve the sequence labelling task. Model hyperparameters were optimised by first defining the search spare and then using Flair's SequenceTaggerParamSelector. Furthermore, the optimal learning rate was found using Flair's find learning rate tool which trains the model starting with a very low learning rate which is then increased exponentially at every batch. The optimal learning rate is the highest value from the optimal phase, found using the graph in figure 1 (Smith, 2017).

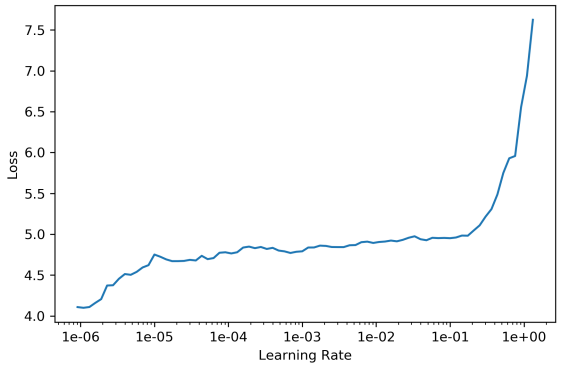


Figure 1: Learning rate for key phrase identification

After optimisation using the development data, the test F1 score key phrase identification was 0.615 on unannotated test data. This is greater than the highest score for subtask A using evaluation scenario 1 for SemEval 2017 Task 10 which was 0.56 achieved using SVM with well-engineered lexical feature set (Augenstein et al., 2018). Another model was created for subtask B which involved the identification of key phrases and subsequent classification which had an F1 score of 0.459. This as well showed superior performance to all entrants to SemEval 2017 Task 10 with the highest score being 0.44 using RNN architecture with CRF layer (Peters et al., 2017).

| Model | Subtask A | Subtask B |
|---|---|---|
| HMM | 0.44 | 0.31 |
| Bi-LSTM-CRF | 0.62 | 0.46 |

Table 3: Micro F1 scores.

## 4 Analysis

The Flair model trained utilises a bidirectional LSTM-CRF architecture and Flair's contextual string embedding. LSTMs contain memory cells which allow long range dependencies in a sentence or paragraph to be captured. In sequence tagging, the past and feature input features at a given time are available so the bidirectional architecture can be utilised. Conditional random field networks make use of neighbouring tokens' tag information by focusing on sentence level tag prediction. Combining Bi-LSTM and CRF has been shown to produce high accuracy tags for sequence labelling (Ma and Hovy, 2016).

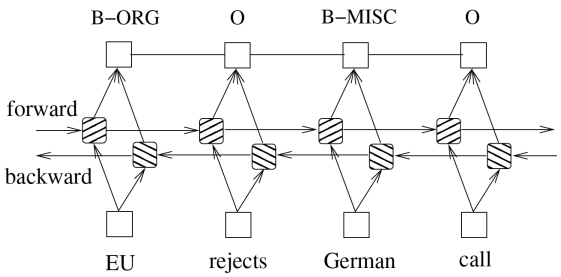


Figure 2: Bi-LSTM-CRF architecture

Flair embeddings are contextual string embeddings which models words as sequences of characters that are contextualised by their surrounding text. The Flair embeddings aim to capture word semantics in context such that words may have different representations depending on the surrounding words (Akbik et al., 2018). The Flair embeddings were concatenated with GloVe embedding to produce representations for sequence tagging. (Akbik et al., 2018) reported using Flair's contextual embedding as part of stacked embeddings achieved better results in a range of NLP tasks.

An example sentence is shown in figure 3 with the gold standard labels showing 4 key phrases and

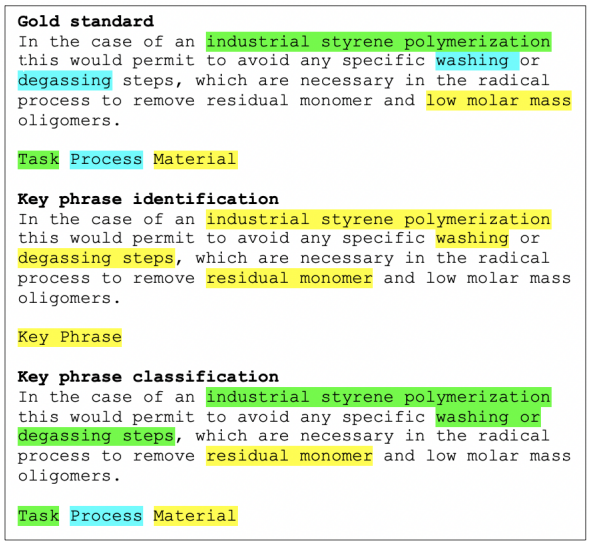


Figure 3: Example text showing gold standard and predicted key phrases

the prediction from the Flair key phrase identification and classification models. The key phrase identification model correctly picks out 3 of the key phrases whilst the classification model shows similar performance in key phrase identification but incorrectly classifies "washing or degassing" as task rather than process.

## 5 Conclusion

The combination of using Flair's contextual string embeddings with Bi-LSTM-CRF architecture outperformed the HMM as well as all entries to SemEval 2017 Task 10 which included neural network architectures for end-to-end learning as well as traditional machine learning such random forests and support vector machine with carefully engineered features. Since the task in 2017 there has been huge advancements in word embeddings. Contextual word representations (used in BERT, ELMo and Flair) produce better representations than static vectors and as a result have been demonstrated to show improvements on a range of NLP tasks. The Flair framework allows these embeddings to be stacked and then incorporated into Bi-LSTM-CRF framework to produce high performing models with relative ease.

The training and optimisation of the Flair models was limited by the available computational power, however this could be overcome by using cloud GPUs. With more GPUs and training time the F1 scores presented here could be exceeded. Furthermore, manual examination of the errors could yield patterns of common errors which could be improved with prepossessing steps and further boost performance of the system.

Identifying and classifying key phrases from scientific publications is a difficult task as key phrases contain specific and infrequent words and can consist of a large number of tokens reducing the generalisation ability. Future work could expand the model to extract semantic relations between key phrases as part of subtask C of SemEval 2017 Task 10.

## References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. *Proceedings of the 27th International Conference on Computational Linguistics*.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2018. SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications.

Daniel Jurafsky and JH Martin. 2000. *Speech & language processing*.

Su Nam Kim, Olena Medelyan, Min Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 Task 5: Automatic keyphrase extraction from scientific articles. In *ACL 2010 - SemEval 2010 - 5th International Workshop on Semantic Evaluation, Proceedings*.

Su Nam Kim, Olena Medelyan, Min Yen Kan, and Timothy Baldwin. 2013. Automatic keyphrase extraction from scientific articles. *Language Resources and Evaluation*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*.

Animesh Prasad and Min-Yen Kan. 2017. WING-NUS at SemEval-2017 Task 10: Keyphrase Extraction and Classification as Joint Sequence Labeling. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Leslie N. Smith. 2017. Cyclical learning rates for training neural networks. In *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*.