Making History Count

Nicolò Verardo

Università degli Studi di Milano, Milano, Italia
nicolo.verardo@studenti.unimi.it
nicoloverardo.surge.sh

Abstract. In this work we try to build an event historical detection system and analyze the different impact that different word embeddings may have on its prediction outcome. We will build from stratch the event detection system and we will train it on a corpus of historical texts. We will compare an historical word embedding with the popular Glove word embedding in order to use it in our model and we will see that, in this case, there is no significant difference. Then, we will test our event detection system on a custom dataset containing abstracts of Wikipedia pages. We will also train a similar model on this custom dataset.

Keywords: event-detection · historical-event-detection · word-embeddings

1 Introduction

A word embedding is a learned representation for text where words that have the same meaning have a similar representation: words that are used in similar ways to result in having similar representations, naturally capturing their meaning. Words are represented as vectors in a vector space and their values are learned in a way that resembles a neural network; this is the reason why they are often used in a deep learning context. Roughly, this translates to represent a word as a distribution over the vocabulary given a corpus of documents. However, different word embeddings may be mapped to different vector spaces: one cannot directly compare two word embeddings but some adjustments are required. Moreover, one can study the differences among word embeddings to learn the evolution of a word since the vectorial representation may capture their meaning.

Event detection from text is one of the many tasks exploiting word embeddings. The complexity of this problem lies in the fact that the definition of "event" is, per se, ambiguous. In addition, there is no unique definition to what a historical event may be, and thus this poses an even more complex problem when building a system that uses deep learning, such as Long Short-Term Memory (LSTM) neural networks.

2 Research question and methodology

This work aims to achieve three milestones: building an event detection system to recognize the presence of historical events on the Histo dataset; comparing a historical and a contemporary word embeddings to study language variations during time; and finally apply the event detection system to Wikipedia pages' abstracts in order to recognize the presence of historical events.

Starting from the Histo dataset, the Glove word embedding and the new Histo word embedding, we built our own BiLSTM + CRF neural network for sequence tagging taking inspiration from [1], [7] and [8]. We trained it both using the historical and contemporary word embedding in order to see if a historical embedding would provide better results, since the dataset may contain terms that are no more in use in the contemporary English.

We then proceeded to compare the two word embeddings. To the best of our knowledge, there is no straightforward technique to compare word embeddings trained on different datasets. Since word embeddings represent words in a different space, firstly we rotated the historical word embedding to the contemporary one in order to get its best approximation possible. Then, we evaluated the distance between vectors representing the same word using the cosine distance [2]; the lower the distance, the more similar the two embeddings are. We chose a set of common and frequently used words for this analysis.

Finally, the last part of this work consisted in downloading Wikipedia webpages by exploiting their API and apply the event detection system to them in order to recognize the presence of historical events. We will see that some problems concerning the ground truth of sentences arose while applying the event detection system to this custom dataset and we will analyze the workaround we have found to this problem.

2.1 The dataset

The Histo dataset contains historical texts of news and travel reports published between the second half of the 19th century and the beginning of the 20th. It is annotated in accordance to specific guidelines published alongside the corpus and it follows the BIO scheme.

CAVA	cava	NN	REPORT	LOCATION	O
DEI	dei	NN	REPORT	LOCATION	O
TIRRENI	tirreni	NN	REPORT	LOCATION	O
,	,	,	REPORT	NONE	O
March	March	NNP	REPORT	NONE	O
8th	8th	JJ	REPORT	NONE	O
.	.	.	REPORT	NONE	O

Table 1: A sentence from the Histo dataset

The BIO notation is a very common format used for Named Entity Recognition. The B- prefix before a tag indicates that the word is the beginning of an annotated segment; the I- prefix before a tag indicates that the word is inside the annotated segment and finally the O tag is used when the word is outside an annotated segment (i.e.: no event to mention).

In total, the Histo dataset contains 22 unique types of annotations. It is already divided into train/dev/test, where we have a respectively 49.896, 6.728 and 6.495 words with the relative tag (notice that punctuation is included but whitespaces are not). Tags (i.e.: labels) are highly unbalanced: the ”outer” tag (”O”) is the majority class with 40.564 appearances. Even not considering the ”B-” or ”I-” prefix, there are tag that appear few times compared to others.

Table 2: Most frequent tags distribution. No IOB scheme on the right

Tag	**Count**
O	40.564
B-SPACEMOVEMENT	1.346
B-COMMUNICATION	807
B-MENTAL	666
B-ACTION	663
I-EMOTIONSEVALUATIONS	550
B-EMOTIONSEVALUATIONS	543
B-EXISTENCECAUSATION	510

Tag	**Count**
O	40.564
SPACEMOVEMENT	1.743
EMOTIONSEVALUATIONS	1093
COMMUNICATION	996
MENTAL	905
ACTION	834
EXISTENCECAUSATION	545
PHYSICALSENSATIONS	539

2.2 Metrics

In order to evaluate the results, we will use mainly the F1-score since our data is very unbalanced and therefore we cannot rely on accuracy. The F1-score is the harmonic mean of precision and recall:

$$\mathrm{F1} = 2 \cdot \frac{(\mathrm{precision} \cdot \mathrm{recall})}{(\mathrm{precision} + \mathrm{recall})} \tag{1}$$

where *precision* can be seen as “the ability of the classifier not to label as positive a sample that is negative”:

$$\mathrm{precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2}$$

and *recall* can be seen as “the ability of the classifier to find all the positive samples”:

$$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

We will use the cosine similarity to compare two vectors representing word embeddings, which is defined as follows:

$$\cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\| \cdot \|\boldsymbol{y}\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} (x_i)^2} \sqrt{\sum_{i=1}^{n} (y_i)^2}} \tag{4}$$

2.3 Recurrent Neural Networks

Recurrent neural networks employ sequential information. They are used to face problems where the ordering of the input sequence is important; in fact, they keep a memory-based on history on information that allows them to predict the current output conditioned on long distance features. The fundamental component of a RNN is a cell. Cells have weights and an internal state. The state is updated by applying the same computation and weights to every element of a sequence, in sequential fashion, over multiple time steps. This state is called "hidden vector" and acts as a "memory". Multiple cells can be stacked forming a multi-layer recurrent neural network. The most popular RNN cell types are long short-term memories (LSTMs). LSTM networks may be better at sequence tagging [3], since in sequence tagging we not only have access to past input features but also to the future ones, so that we can make a LSTM that analyzes the sequence both in forward and backward direction (i.e.: bidirectional LSTM).

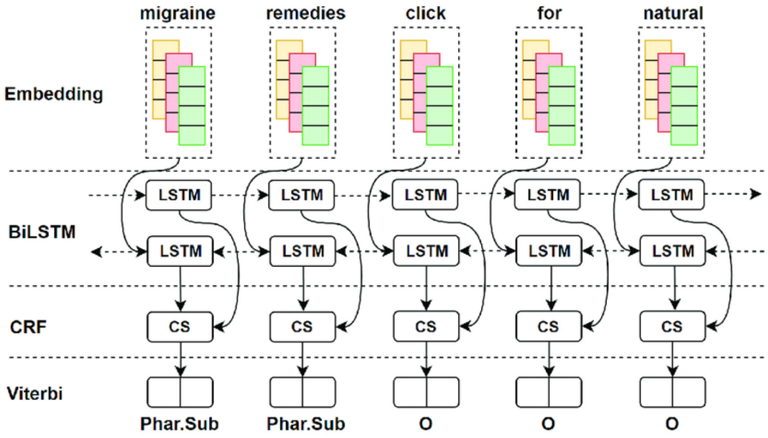
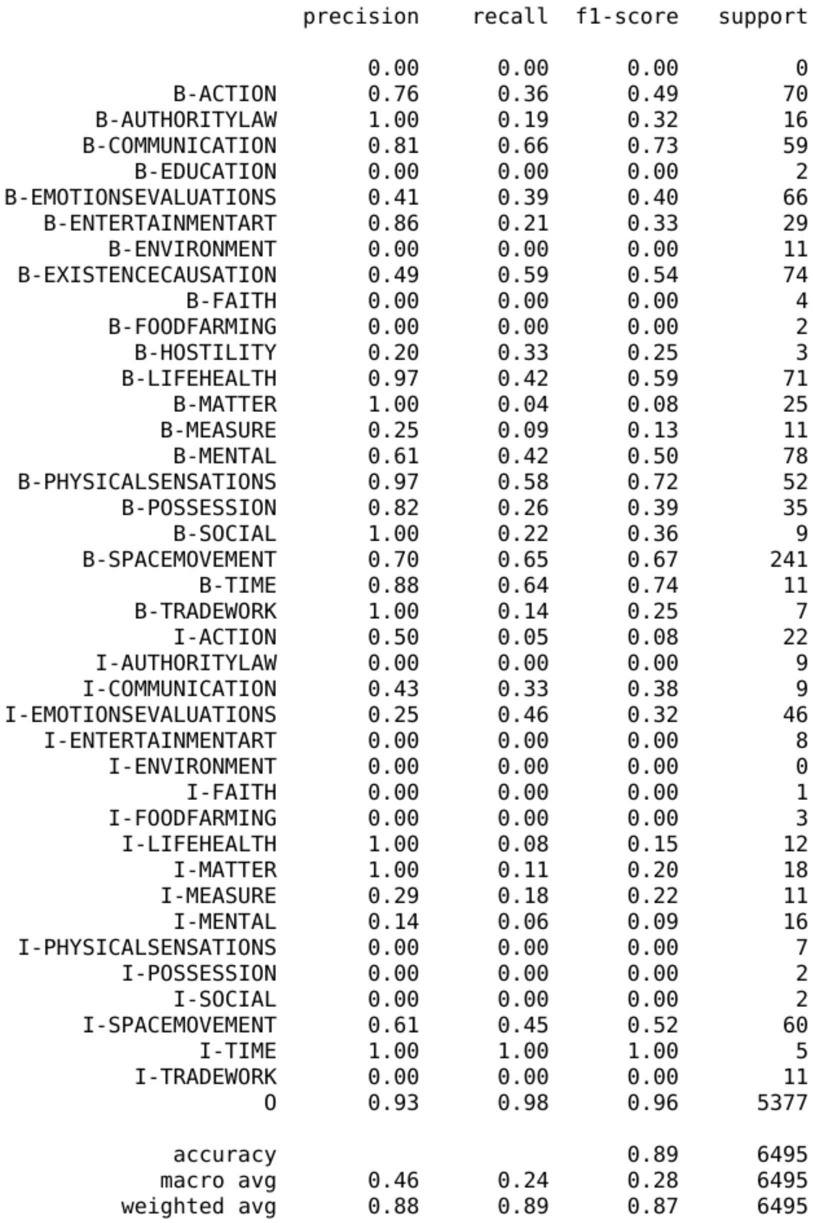


Fig. 1: Architecture of a BiLSTM + CRF network

3 Experimental results

The experiments were run on a free Google Colab instance running Ubuntu that provided a quad-core Intel Xeon E5-2650 v3 that run at 2,30 Mhz and 12GB of RAM. Moreover, it provided also a NVIDIA Tesla K80 as GPU that we used to train the BiLSTM + CRF using `keras` with `TensorFlow` 2.3 as backend.

In order to correctly detect tags of words of the Histo dataset, we built the event detection system from scratch. Following [1], we built a BiLSTM neural network that has a CRF layer stacked as output. Similar models, such as [1] and [10], are already available online and almost ready to use (with few modifications) on the Histo dataset; however, most of them are built on TensorFlow 1.x and their implementation is rather complex. Thus, we wanted to build a more simple and straightforward model using keras that is compatible with TensorFlow 2.x.
The input (that is a list of vectorized sentences into unique token indices) is embedded using either Glove or the historical embeddings to 300 dimensions and a dropout of 0.5 is applied. Next, two bidirectional LSTM layers with default size of 128 are stacked before a time distributed layer, which applies a dense layer to each word of the sentence. Then, we apply another dropout of size 0.5 to fight overfitting and finally a CRF layer is stacked and it will output the predicted tag for each word. We used a batch size of 32 for 100 epochs, but applying early stopping when the loss stopped decreasing significantly.
We trained two models: one using the Glove embeddings and one using the historical embeddings. Their results were very similar, both in terms of F1 score, and in terms of the ability to spot and predict all tags of words. In fact, as we can see in Fig. 2, there are classes whose score is zero: because of the low number of examples, the model was not able to detect them. We got F1 scores of 87% and 86% using the Glove and historical embeddings respectively.



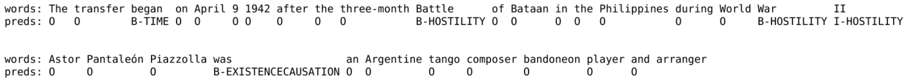
```
                        precision    recall  f1-score   support

                             0.00      0.00      0.00         0
             B-ACTION        0.76      0.36      0.49        70
       B-AUTHORITYLAW        1.00      0.19      0.32        16
      B-COMMUNICATION        0.81      0.66      0.73        59
          B-EDUCATION        0.00      0.00      0.00         2
B-EMOTIONSEVALUATIONS        0.41      0.39      0.40        66
   B-ENTERTAINMENTART        0.86      0.21      0.33        29
        B-ENVIRONMENT        0.00      0.00      0.00        11
 B-EXISTENCECAUSATION        0.49      0.59      0.54        74
              B-FAITH        0.00      0.00      0.00         4
        B-FOODFARMING        0.00      0.00      0.00         2
          B-HOSTILITY        0.20      0.33      0.25         3
         B-LIFEHEALTH        0.97      0.42      0.59        71
             B-MATTER        1.00      0.04      0.08        25
            B-MEASURE        0.25      0.09      0.13        11
             B-MENTAL        0.61      0.42      0.50        78
 B-PHYSICALSENSATIONS        0.97      0.58      0.72        52
         B-POSSESSION        0.82      0.26      0.39        35
             B-SOCIAL        1.00      0.22      0.36         9
      B-SPACEMOVEMENT        0.70      0.65      0.67       241
               B-TIME        0.88      0.64      0.74        11
          B-TRADEWORK        1.00      0.14      0.25         7
             I-ACTION        0.50      0.05      0.08        22
       I-AUTHORITYLAW        0.00      0.00      0.00         9
      I-COMMUNICATION        0.43      0.33      0.38         9
I-EMOTIONSEVALUATIONS        0.25      0.46      0.32        46
   I-ENTERTAINMENTART        0.00      0.00      0.00         8
        I-ENVIRONMENT        0.00      0.00      0.00         0
              I-FAITH        0.00      0.00      0.00         1
        I-FOODFARMING        0.00      0.00      0.00         3
         I-LIFEHEALTH        1.00      0.08      0.15        12
             I-MATTER        1.00      0.11      0.20        18
            I-MEASURE        0.29      0.18      0.22        11
             I-MENTAL        0.14      0.06      0.09        16
 I-PHYSICALSENSATIONS        0.00      0.00      0.00         7
         I-POSSESSION        0.00      0.00      0.00         2
             I-SOCIAL        0.00      0.00      0.00         2
      I-SPACEMOVEMENT        0.61      0.45      0.52        60
               I-TIME        1.00      1.00      1.00         5
          I-TRADEWORK        0.00      0.00      0.00        11
                    O        0.93      0.98      0.96      5377

             accuracy                            0.89      6495
            macro avg        0.46      0.24      0.28      6495
         weighted avg        0.88      0.89      0.87      6495
```

Fig. 2: Results on the test set using the Glove embedding

When testing the model on two sentences, we can see that it is able to classify entities of various kind. Among these, it can spot entities that refer to historical events correctly, while still making some mistakes.



```
words: The transfer began  on April 9 1942 after the three-month Battle      of Bataan in the Philippines during World War         II
preds: O   O        B-TIME O  O     O O    O     O   O           B-HOSTILITY O  O      O  O   O           O      O     B-HOSTILITY I-HOSTILITY


words: Astor Pantaleón Piazzolla was                  an Argentine tango composer bandoneon player and arranger
preds: O     O         O         B-EXISTENCECAUSATION O  O         O     O        O         O      O
```

Fig. 3: Sentence prediction of entities

For the second task, namely the comparison of word embeddings, we rotated the historical embedding in order to make it closer to the glove embedding. This happens because different word embeddings may not use the same space to represent words. Then, since words are represented with vectors of dimension 300, we can compare the distance among the two embeddings. The reason behind this is that the relative position of words across embeddings should be similar.

Table 3: Comparison of cosine distance in the embeddings

Word	**Histo - Glove**	**Rot. Histo - Glove**
man	0.972540	0.201267
woman	1.028170	0.155820
boy	0.988347	0.180521
girl	1.016846	0.170501
father	1.013273	0.180857
mother	1.068560	0.180408
male	1.050831	0.300703
female	1.016651	0.292890

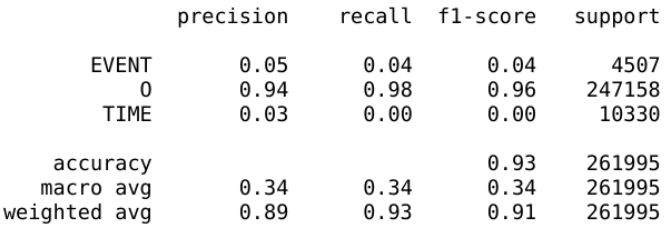
Table 3 clearly shows that the adjustment has reduced the discrepancy between the two embeddings. Nonetheless, results from the previous point have induced us to use the Glove embedding for the last task.

Table 4: Comparison of opposite words distance

Words	**Glove**	**Histo**	**Rotated Histo**	**Glove - Rotated Histo**
man - woman	0.259826	0.337892	0.337892	0.387624
boy - girl	0.185168	0.291651	0.291651	0.294225
father - mother	0.170175	0.238817	0.238817	0.302326
male - female	0.065723	0.198832	0.198832	0.316567

In the above table we can see that opposite words (man - woman) are closer in the Glove embedding rather than in the historical one. Even after the adjustment, the distance among opposite words has not changed much in the historical embedding, thus remaining higher than the ones reported using Glove.

For what concerns the third and last task, we created a custom small dataset of Wikipedia pages and we applied the event detection model to it.
First of all, we retrieved the title and the link of 1.000 pages that were tagged as “MilitaryConflict”, 500 pages tagged “Artist” and 500 tagged as “Animal” by performing three distinct queries using sparql on the dbpedia endpoint. We labelled as 1 pages pertaining historical events (i.e.: MilitaryConflict tag), 0 otherwise (i.e.: Artist and Animal tags). Next, we used the unique title to download the abstract of the pages through the official Wikipedia API. We skipped disambiguations: if we could not find the abstract corresponding to the title or if it was pointing to two different pages, then we simply skipped it leaving the abstract blank and subsequently dropping the corresponding row in our dataset. We end up with 1791 pages, 830 of which are labelled as 1 and 961 labelled as 0.
Although we exploited ontologies to give a ground truth (”historical event” = 1 or not = 0) to sentences, this labelling was not suited in order to apply the historical event detection system, since it required labels in the IOB format. Thus, we resorted to spaCy and performed NER on our Wikipedia dataset in order to use the entities spaCy predicted as ground truth. We mapped both spaCy and our BiLSTM predictions to the same three labels: ”TIME”, ”EVENT”, ”O”.
The rationale behind this being the lack of a definition of a historical event: we have no rule(s) that tell us what features allow us to predict if an entity is a historical event or not. Thus, we resorted to this simplification: words tagged as ”HOSTILITY” or ”EXISTENCECAUSATION” are considered to be under ”EVENT” tag, ”TIME” remains as it is; all the other tags go under ”O”. Similar reasoning is applied to the spaCy predictions: ”DATE” and ”TIME” go under ”TIME”, ”EVENT” stays the same all the other tags go under ”O”. We will see that this is a very rough and not very accurate mapping, although it gives us a glimpse a glimpse at evaluating if our model may work in presence of more appropriate rules. In fact, results show that the majority of the tags correctly predicted are of type ”O”, and the other two tags have almost zero F1 score.



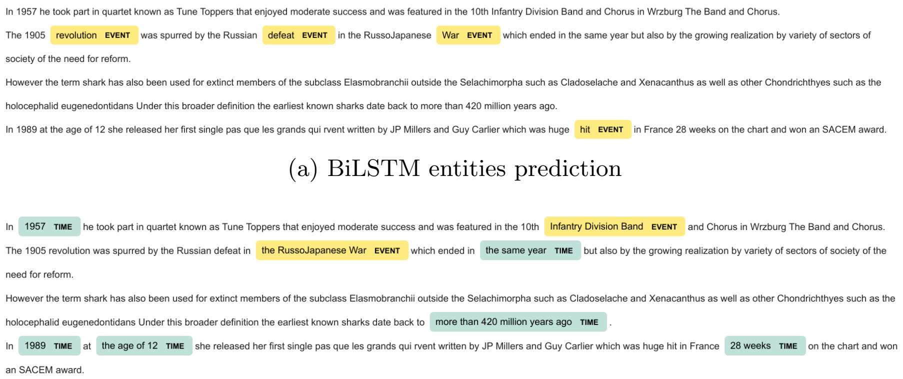
```
              precision    recall  f1-score   support

       EVENT       0.05      0.04      0.04      4507
           O       0.94      0.98      0.96    247158
        TIME       0.03      0.00      0.00     10330

    accuracy                           0.93    261995
   macro avg       0.34      0.34      0.34    261995
weighted avg       0.89      0.93      0.91    261995
```

Fig. 4: Results of our BiLSTM applied to the custom Wikipedia dataset

Displaying randomly four sentences and their prediction, we can outline some differences. In figure 5 we can see the predictions made using our BiLSTM model and those made using spaCy NER.
Our model is able to recognize as events words like ”War”, but not the entire noun chunk ”Russo-Japanese War”; it also recognizes words like ”revolution” and ”defeat” as events, while they may not directly refer to a historical event. It misses dates or timestamps like 1957 and ”more than 420 million years ago”, which are instead recognized correctly by spaCy (although it missed ”1905”). However, our model correctly skips (i.e.: classify as ”O”) words that belong to sentences not pertinent to a historical event: in the first sentence, concerning music, spaCy classifies as event ”Infantry Division Band”, while our model correctly classifies it as ”O”.



In 1957 he took part in quartet known as Tune Toppers that enjoyed moderate success and was featured in the 10th Infantry Division Band and Chorus in Wrzburg The Band and Chorus.

The 1905 revolution EVENT was spurred by the Russian defeat EVENT in the RussoJapanese War EVENT which ended in the same year but also by the growing realization by variety of sectors of society of the need for reform.

However the term shark has also been used for extinct members of the subclass Elasmobranchii outside the Selachimorpha such as Cladoselache and Xenacanthus as well as other Chondrichthyes such as the holocephalid eugenedontidans Under this broader definition the earliest known sharks date back to more than 420 million years ago.

In 1989 at the age of 12 she released her first single pas que les grands qui rvent written by JP Millers and Guy Carlier which was huge hit EVENT in France 28 weeks on the chart and won an SACEM award.

(a) BiLSTM entities prediction

In 1957 TIME he took part in quartet known as Tune Toppers that enjoyed moderate success and was featured in the 10th Infantry Division Band EVENT and Chorus in Wrzburg The Band and Chorus.

The 1905 revolution was spurred by the Russian defeat in the RussoJapanese War EVENT which ended in the same year TIME but also by the growing realization by variety of sectors of society of the need for reform.

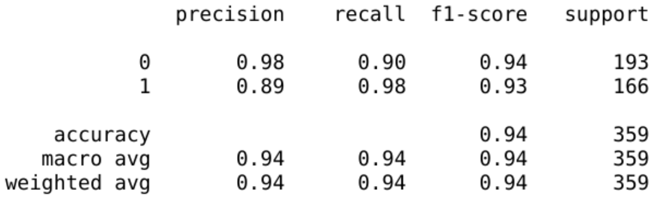
However the term shark has also been used for extinct members of the subclass Elasmobranchii outside the Selachimorpha such as Cladoselache and Xenacanthus as well as other Chondrichthyes such as the holocephalid eugenedontidans Under this broader definition the earliest known sharks date back to more than 420 million years ago TIME .

In 1989 TIME at the age of 12 TIME she released her first single pas que les grands qui rvent written by JP Millers and Guy Carlier which was huge hit in France 28 weeks TIME on the chart and won an SACEM award.

(b) spaCy entities prediction

Fig. 5: Comparison of entities prediction

Moreover, also wanted to train a model similar to the BiLSTM that worked with the original binary classification of Wikipedia abstracts (even though we do not have a massive amount of pages). We proceeded similarly as in the BiLSTM model: we used the same embedding layer with Glove, two bidirectional LSTM layers with size 128, but then we directly attached a dense layer with 128 neurons and relu activation, a dropout of 0.5 and finally the output layer.
With a batch size of 64, we achieved a F1 score of 94% in just 4 epochs. Results are shown in Fig. 6.



```
              precision    recall  f1-score   support

           0       0.98      0.90      0.94       193
           1       0.89      0.98      0.93       166

    accuracy                           0.94       359
   macro avg       0.94      0.94      0.94       359
weighted avg       0.94      0.94      0.94       359
```

Fig. 6: Binary classification on the Wikipedia dataset

These results show that, with the original binary classification, we are able to distinguish between texts that contain historical events and texts that do not.

4 Concluding remarks

The analysis of word embeddings shows that opposite words (man/woman) are less distant in the Glove embedding, rather than in the historical one. This may suggest that the two embeddings would give different results when applied to the event detection system. However, this is not true as shown in the previous section, since we got F1 scores of 87% and 86% using the Glove and historical embeddings respectively; this led us to use the Glove embedding on the Wikipedia dataset, since the historical embedding did not provide any significant improvement to the results.
Moreover, the analysis of the two embedding suggests that, even though there may be differences, they are not that relevant for the purpose of event detection.
Finally, one can employ different techniques that would provide a better comparison of word embeddings, like comparing distances of neighbouring words by exploiting, for instance, the Nearest Neighbors approach implemented in [9].

Results of the event detection system show that our model may be suited for NER tasks on datasets with BIO format, but when applied to the Wikipedia dataset results are not satisfying: we may have reached a high F1 score but, as we have pointed out in the previous section, we are not able to spot all the tags nor to reach the goal of detecting historical events. The cause can be mainly found into two aspects of the work: the first being the mapping of the tags, and the second being the lack of rules to correctly define what is a historical event.

In conclusion, a big improvement of this work would be training the model using different corpora and applying it to a dataset that uses BIO format and then output the results using a wiser mapping than the one used in this work.

References

1. Huang, Xu, Yu. Bidirectional LSTM-CRF Models for Sequence Tagging. arXiv:1508.0199. 2015
2. Yanshan Wang, Sijia Liu, Naveed Afzal, MajidRastegar-Mojarad, Liwei Wang, Feichen Shen, PaulKingsbury, and Hongfang Liu. 2018b. A comparison ofword embeddings for the biomedical natural languageprocessing. Journal of biomedical informatics 87 (2018), 12–20.
3. Thakker, Dasika, Beu and Mattina. Measuring scheduling efficiency of RNNs for NLPapplications. arXiv:1904.03302v1. 2019
4. Trevor Hastie, Robert Tibshirani, Jerome Friedman. The Elements of Statistical Learning. Springer Series in Statistics, 2017.
5. Shalev-Shwartz, Ben-David - Understanding Machine Learning. Cambridge University Press, 2014.
6. https://github.com/dhfbk/Histo
7. https://www.depends-on-the-definition.com/sequence-tagging-lstm-crf/
8. https://github.com/guillaumegenthial/tf_ner
9. https://pypi.org/project/repcomp/
10. https://aihub.cloud.google.com/p/products%2F2290fc65-0041-4c87-a898-0289f59aa8ba