WBI at CLEF eHealth 2018 Task 1: Language-independent ICD-10 coding using multi-lingual embeddings and recurrent neural networks

Jurica Ševa, Mario Sänger, and Ulf Leser

Humboldt-Universität zu Berlin, Knowledge Management in Bioinformatics,
Berlin, Germany
{seva,saengema,leser}@informatik.hu-berlin.de

Abstract. This paper describes the participation of the WBI team in the CLEF eHealth 2018 shared task 1 ("Multilingual Information Extraction - ICD-10 coding"). Our contribution focus on the setup and evaluation of a baseline language-independent neural architecture for ICD-10 classification as well as a simple, heuristic multi-language word embedding space. The approach builds on two recurrent neural networks models to extract and classify causes of death from French, Italian and Hungarian death certificates. First, we employ a LSTM-based sequence-to-sequence model to obtain a death cause from each death certificate line. We then utilize a bidirectional LSTM model with attention mechanism to assign the respective ICD-10 codes to the received death cause description. Both models take multi-language word embeddings as inputs. During evaluation our best model achieves an F-score of 0.34 for French, 0.45 for Hungarian and 0.77 for Italian. The results are encouraging for future work as well as the extension and improvement of the proposed baseline system.

Keywords: ICD-10 coding · Biomedical information extraction · Multilingual sequence-to-sequence model · Represention learning · Recurrent neural network · Attention mechanism · Multi-language embeddings

1 Introduction

Automatic extraction, classification and analysis of biological and medical concepts from unstructured texts, such as scientific publications or electronic health documents, is a highly important task to support many applications in research, daily clinical routine and policy-making. Computer-assisted approaches can improve decision making and support clinical processes, for example, by giving a more sophisticated overview about a research area, providing detailed information about the aetiopathology of a patient or disease patterns. In the past years major advances have been made in the area of natural-language processing (NLP). However, improvements in the field of biomedical text mining lag behind other domains mainly due to privacy issues and concerns regarding the processed data (e.g. electronic health records).

The CLEF eHealth lab[1] attends to circumvent this situation through organization of various shared tasks to exploit electronically available medical content [34]. In particular, Task 1[2] of the lab is concerned with the extraction and classification of death causes from death certificates originating from different languages [27]. Participants were asked to classify the death causes mentioned in the certificates according to the International Classification of Disease version 10 (ICD-10)[3]. The task was concerned with French and English death certificates in previous years. In contrast, this year the organizers provided annotated death reports as well as ICD-10 dictionaries for French, Italian and Hungarian. The development of language-independent, multilingual approaches was encouraged.

Inspired by the recent success of recurrent neural network models (RNN) [6,20,10] in general and the convincing performance of the work from Miftahutdinov and Tutubalina [21] in the last edition of the lab, we opt for the development of a deep learning model for this year's competition. Our work introduces a prototypical, language independent approach for ICD-10 classification using multi-language word embeddings and long short-term memory models (LSTMs). We divide the proposed pipeline into two tasks. First, we perform named entity recognition (NER), i.e. extract the death cause description from a certificate line, with an an encoder-decoder model. Given the death cause, named entity normalization (NEN), i.e. assigning an ICD-10 code to extracted death cause, is performed by a separate LSTM. Our approach builds upon a heuristic multi-language embedding space and therefore only needs one single model for all three data sets. With this work we want to experiment and evaluate which performance can be achieved with such a simple shared embedding space.

2 Related work

This section highlights previous work related to our approach. We give a brief introduction to the methodical foundations of our work, RNNs and word embeddings. The section concludes with a summary of ICD-10 classification approaches used in previous eHealth Lab competitions.

2.1 Recurrent neural networks (RNN)

RNNs are a widely used technique for sequence learning problems such as machine translation [1,6], image captioning [2], NER [20,40], dependency parsing [10] and part-of-speech tagging [39]. RNNs model dynamic temporal behaviour in sequential data through recurrent units, i.e. the hidden, internal state of a unit in one time step depends on the state of the unit in the previous time step.

[1] https://sites.google.com/site/clefehealth/

[2] https://sites.google.com/view/clef-ehealth-2018/task-1-multilingual-information-extraction-icd10-coding

[3] http://www.who.int/classifications/icd/en/

These feedback connections enable the network to memorize information from recent time steps and add the ability to capture long-term dependencies.

However, training of RNNs can be difficult due to the vanishing gradient problem [16,3]. The most widespread modifications of RNNs to overcome this problem are LSTMs [17] and gated recurrent units (GRU) [6]. Both modifications use gated memories which control and regulate the information flow between two recurrent units. A common LSTM unit consists of a cell and three gates, an input gate, an output gate and a forget gate. In general, LSTMs are chained together by connecting the outputs of the previous unit to the inputs of the next one.

A further extension of the general RNN architecture are bidirectional networks, which make the past and future context available in every time step. A bidirectional LSTM model consists of a forward chain, which processes the input data from left to right, and and backward chain, consuming the data in the opposite direction. The final representation is typically the concatenation or a linear combination of both states.

2.2 Word Embeddings

Distributional semantic models (DSMs) have been researched for decades in NLP [37]. Based on a huge amount of unlabeled texts, DSMs aim to represent words using a real-valued vector (also called embedding) which captures syntactic and semantic similarities between the words. Starting with the publication of the work from Collobert et al. [7] in 2011, learning embeddings for linguistic units, such as words, sentences or paragraphs, is one of the hot topics in NLP and a plethora of approaches have been proposed [4,23,28,30].

The majority of todays embedding models are based on deep learning models trained to perform some kind of language modeling task [29,30,32]. The most popular embedding model is the Word2Vec model introduced by Mikolov et al. [22,23]. They propose two shallow neural network models, continuous bag-of-words (CBOW) and SkipGram, that are trained to reconstruct the context given a center word and vice versa. In contrast, Pennington et al. [28] use the ratio between co-occurrence probabilities of two words with another one to learn a vector representation. In [30] multi-layer, bi-directional LSTM models are utilized to learn word embeddings that also capture different contexts of it.

Several recent models focus on the integration of subword and morphological information to provide suitable representations even for unseen, out-of-vocabulary words. For example, Pinter et al. [32] try to reconstruct a pre-trained word embedding by learning a bi-directional LSTM model on character level. Similarly, Bojanowski et al. [4] adapt the SkipGram by taking character n-grams into account. Their fastText model assigns a vector representation to each character n-gram and represents words by summing over all of these representations of a word.

In addition to embeddings that capture word similarities in one language, multi- and cross-lingual approaches have also been investigated. Proposed methods either learn a linear mapping between monolingual representations [12,41] or utilize word- [13,38], sentence- [31] or document-aligned [36] corpora to build a shared embedding space.

2.3 ICD-10 Classification

The ICD-10 coding task has already been carried out in the 2016 [26] and 2017 [25] edition of the eHealth lab. Participating teams used a plethora of different approaches to tackle the classification problem. The methods can essentially be divided into two categories: knowledge-based [5,18,24] and machine learning (ML) approaches [8,11,15,21]. The former relies on lexical sources, medical terminologies and other dictionaries to match (parts of) the certificate text with entries from the knowledge-bases according to a rule framework. For example, Di Nunzio et al. [9] calculate a score for each ICD-10 dictionary entry by summing the binary or tf-idf weights of each term of a certificate line segment and assign the ICD-10 code with the highest score. In contrast, Ho-Dac et al. [14] treat the problem as information retrieval task and utilize the Apache Solr search engine[4] to classify the individual lines.

The ML-based approaches employ a variety of techniques, e.g. Conditional Random Fields (CRFs) [15], Labeled Latent Dirichlet Analysis (LDA) [8] and Support Vector Machines (SVMs) [11] with diverse hand-crafted features. Most similar to our approach is the work from Miftahutdinov and Tutubalina [21], which achieved the best results for English certificates in the last year's competition. They use a neural LSTM-based encoder-decoder model that processes the raw certificate text as input and encodes it into a vector representation. Additionally, a vector which captures the textual similarity between the certificate line and the death causes of the individual ICD-10 codes is used to integrate prior knowledge into the model. The concatenation of both vector representations is then used to output the characters and numbers of the ICD-10 code in the decoding step. In contrast to their work, our approach introduces a model for multi-language ICD-10 classification. Moreover, we divide the task into two distinct steps: death cause extraction and ICD-10 classification.

3 Methods

Our approach models the extraction and classification of death causes as two-step process. First, we employ a neural, multi-language sequence-to-sequence model to receive a death cause description for a given death certificate line. We then use a second classification model to assign the respective ICD-10 codes to the obtained death cause. The remainder of this section gives a detailed explanation of the architecture of the two models.

3.1 Death Cause Extraction Model

The first step in our pipeline is the extraction of the death cause from a given certificate line. We use the training certificate lines (with their corresponding

[4] http://lucene.apache.org/solr/

ICD-10 codes) and the ICD-10 dictionaries as basis for our model. The dictionaries provide us with death causes for each ICD-10 code. The goal of the model is to reassemble the dictionary death cause text from the certificate line.

For this we adopt the encoder-decoder architecture proposed in [35]. Figure 1 illustrates the architecture of the model. As encoder we utilize a unidirectional LSTM model, which takes the single words of a certificate line as inputs and scans the line from left to right. Each token is represented using pre-trained fastText[5] word embeddings [4]. We utilize fastText embedding models for French, Italian and Hungarian trained on Common Crawl and Wikipedia articles[6]. Independently from the original language a word we represent it by looking up the word in all three embedding models and concatenate the obtained vectors. Through this we get a simple multi-language representation of the word. This heuristic composition constitutes a naive solution to build a multi-language embedding space. However we opted to evaluate this approach as a simple baseline for future work. Encoders' final state represents the semantic representation of the certificate line and serves as initial input for decoding process.

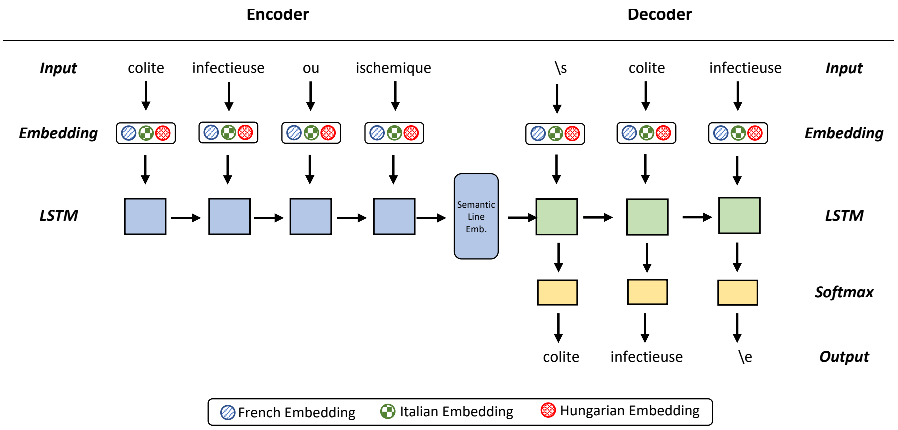

Fig. 1. Illustration of the encoder-decoder model for death cause extraction. The encoder processes a death certificate line token-wise from left to right. The final state of the encoder forms a semantic representation of the line and serves as initial input for the decoding process. The decoder will be trained to predict the death cause text from the provided ICD-10 dictionaries word by word (using special tags \s and \e for start resp. end of a sequence). All input tokens will be represented using the concatenation of the fastText embeddings of all three languages.

For the decoder we utilize another LSTM model. The initial input of the decoder is the final state of the encoder model. Moreover, each token of the

[5] https://github.com/facebookresearch/fastText/

[6] https://github.com/facebookresearch/fastText/blob/master/docs/crawl-vectors.md

dictionary death cause text (padded with special start and end tag) serves as (sequential) input. Again, we use fastText embeddings of all three languages to represent the input tokens. The decoder predicts one-hot-encoded words of the death cause. During test time we use the encoder to obtain a semantic representation of the certificate line and decode the death cause description word by word starting with the special start tag. The decoding process finishes when the decoder outputs the end tag.

3.2 ICD-10 Classification Model

The second step in our pipeline is to assign a ICD-10 code to the generated death cause description. For this we employ a bidirectional LSTM model which is able to capture the past and future context for each token of a death cause description. Just as in our encoder-decoder model we encode each token using the concatenation of the fastText embeddings of the word from all three languages. To enable our model to attend to different parts of the death cause we add an extra attention layer [33] to the model. Through the attention mechanism our model learns a fixed-sized embedding of the death cause description by computing an adaptive weighted average of the state sequence of the LSTM model. This allows the model to better integrate information over time. Figure 2 presents the architecture of our ICD-10 classification model. We train the model using the provided ICD-10 dictionaries from all three languages.

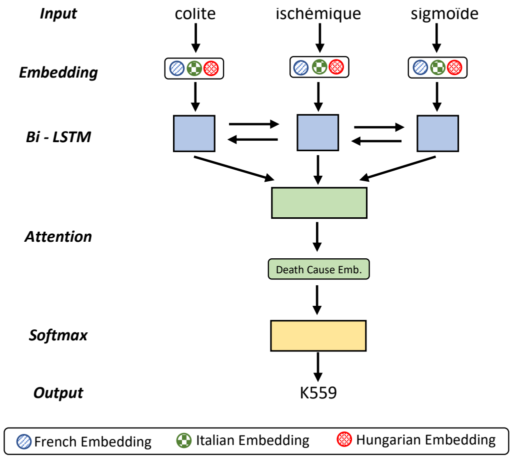

Fig. 2. Illustration of the ICD-10 classification model. The model utilizes a bi-directional LSTM layer, which processes the death cause from left to right and vice versa. The attention layer summarizes the whole description by computing an adaptive weighted average over the LSTM states. The resulting death cause embedding will be feed through a softmax layer to get the final classification. Equivalent to our encoder-decoder model all input tokens will be represented using the concatenation of the fastText embeddings of all three languages.

4 Experiments and Results

In this section we will present experiments and obtained results for the two developed models, both individually as well as combined in a pipeline setting.

4.1 Training Data and Experiment Setup

The CLEF e-Health 2018 Task 1 participants where provided with annotated death certificates for the three selected languages: French, Italian and Hungarian. Each of the languages is supported by training certificate lines as well as a dictionary with death cause descriptions resp. diagnosis for the different ICD-10 codes. The provided training data sets were imbalanced concerning the different languages: the Italian corpora consists of 49,823, French corpora of 77,348[7] and Hungarian corpora 323,175 certificate lines. We split each data set into a training and a hold-out evaluation set. The complete training data set was then created by combining the certificate lines of all three languages into one data set. Beside the provided certificate data we used no additional knowledge resources or annotated texts.

Due to time constraints during development no cross-validation to optimize the (hyper-) parameters and the individual layers of our models was performed. We either keep the default values of the hyper-parameters or set them to reasonable values according to existing work. During model training we shuffle the training instances and use varying instances to perform a validation of the epoch.

Pre-trained fastText word embeddings were trained using the following parameter settings: CBOW with position-weights, embedding dimension size 300, with character n-grams of length 5, a window of size 5 and 10 negative samples. Unfortunately, they are trained on corpora not related with the biomedical domain and therefore do not represent the best possible textual basis for an embedding space for biomedical information extraction. Final embedding space used by our models is created by concatenating individual embedding vectors for all three languages. Thus the input of our model is embedding vector of size 900. All models were implemented with the Keras[8] library.

4.2 Death cause extraction model

To identify possible candidates for a death cause description, we focus on the use of an encoder-decoder model. The encoder model uses an embedding layer with input masking on zero values and a LSTM layer with 256 units. The encoders' output is used as the initial state of the decoder model.

Based on the input description from the dictionary and a special start token, the decoder generates a death cause word by word. This decoding process continues until a special end token is generated. The entire model is optimized using the Adam optimization algorithm [19] and a batch size of 700. Model training

[7] For French we only took the provided data set from 2014.

[8] https://keras.io/

was performed either for 100 epochs or until an early stopping criteria is met (no change in validation loss for two epochs).

As the provided data set are imbalanced regarding the tasks' languages, we devised two different evaluation settings: (1) DCEM-Balanced, where each language was supported by 49.823 randomly drawn instances (size of the smallest corpus) and (2) DCEM-Full, where all available data is used. Table 4.2 shows the results obtained on the training and validation set. The figures indicate that the distribution of training instances per language have a huge influence on the performance of the model. The model trained on the full training data achieves an accuracy of 0.678 on the validation set. In contrast using the balanced data set the model reaches an accuracy of 0.899 (+ 32.5%).

Setting	**Trained Epochs**	**Train**		**Validation**	
		Accuracy	**Loss**	**Accuracy**	**Loss**
DCEM-Balanced	18	0.958	0.205	0.899	0.634
DCEM-Full	9	0.709	0.098	0.678	0.330

Table 1. Experiment results of our death cause extraction sequence-to-sequence model concerning balanced (equal number of training instances per language) and full data set setting.

4.3 ICD-10 Classification Model

The classification model is responsible for assigning a ICD-10 code to death cause description obtained during the first step. Our model uses an embedding layer with input masking on zero values, followed by a bidirectional LSTM layer with 256 dimension hidden layer. Thereafter an attention layer builds an adaptive weighted average over all LSTM states. The respective ICD-10 code will be determined by a dense layer with softmax activation function. We use the Adam optimizer to perform model training. The model was validated on 25% of the data. As for the extraction model, no cross-validation or hyper-parameter optimization was performed.

Once again, we devised two approaches. This was mainly caused by the lack of adequate training data in terms of coverage for individual ICD-10 codes. Therefore, we defined two training data settings: (1) minimal (ICD-10_Minimal), where only ICD-10 codes with two or more supporting training instances are used. This leaves us with 6,857 unique ICD-10 codes and discards 2,238 unique codes with support of one. This, of course, minimizes the number of ICD-10 codes in the label space. Therefore, (2) an extended (ICD-10_Extended) data set was defined. Here, the original ICD-10 code mappings, found in the supplied dictionaries, are extended with the training instances from individual certificate data from the three languages. This generates 9,591 unique ICD-10 codes. Finally, for the remaining ICD-10 codes that have only one supporting description, we duplicate those data points.

The goal of this approach is to extend our possible label space to all available ICD-10 codes. The results obtained from the two approaches on the validation set are shown in Table 4.3. Using the *minimal* data set the model achieves an accuracy of 0.937. In contrast, using the extended data set the model reaches an accuracy of 0.954 which represents an improvement of 1.8%.

Setting	**Trained Epochs**	**Train**		**Validation**	
		Accuracy	**Loss**	**Accuracy**	**Loss**
ICD-10_Minimal	69	0.925	0.190	0.937	0.169
ICD-10_Extended*****	41	0.950	0.156	0.954	0.141

Table 2. Experiment results for our ICD-10 classification model regarding different data settings. The *Minimal* setting uses only ICD-10 codes with two or more training instances in the supplied dictionary. In contrast, *Extended* additionally takes the diagnosis texts from the certificate data and duplicates ICD-10 training instances with only one diagnosis text in the dictionary and certificate lines. ***** Used in final pipeline.

4.4 Complete Pipeline

The two models where combined to create the final pipeline. We tested both death cause extraction models (based on the balanced and unbalanced data set) in the final pipeline, as their performance differs greatly. On the contrary, both ICD-10 classification models perform similarly, so we just used the extended ICD-10 classification model, with word level tokens[9], in the final pipeline. To evaluate the pipeline we build a training and a hold-out validation set during development. The obtained results on the validation set are presented in Table 4.4. The scores are calculated using a prevalence-weighted macro-average across the output classes, i.e. we calculated precision, recall and F-score for each ICD-10 code and build the average by weighting the scores by the number occurrences of the code in the gold standard.

Although the individual models, as shown in Tables 4.2 and 4.3 are promising, the performance decreases considerably in a pipeline setting . The pipeline model based on the balanced data set reaches a F-score of 0.61, whereas the full model achieves a slightly higher value of 0.63. Both model configurations have a higher precision than recall (0.73/0.61 resp. 0.74/0.62).

This can be contributed to several factors. First of all, a pipeline architecture always suffers from error-propagation, i.e. errors in a previous step will influence the performance of the following layers and generally lower the performance of the overall system. Investigating the obtained results, we found that the imbalanced distribution of ICD-10 codes represents one the main problems. This

[9] Although models supporting character level tokens were developed and evaluated, their performance fared poorly compared to the word level tokens.

Model	**Precision**	**Recall**	**F-score**
Final-Balanced	0.73	0.61	0.61
Final-Full	0.74	0.62	0.63

Table 3. Evaluation results of the final pipeline on the validation set of the training data. Reported figures represent the prevalence-weighted macro-average across the output classes. Final-Balanced = DCEM-Balanced + ICD-10_Extended. Final-Full = DCEM-Full + ICD-10_Extended

severely impacts the decoder-encoder architecture used here as the token generation is biased towards the available data points. Therefore the models misclassify certificate lines associated with ICD-10 codes that only have a small number of supporting training instances very often.

Results obtained on the test data set, resulting from the two submitted official runs, are shown in Table 4. Similar to the evaluation results during development, the model based on the full data set performs slightly better than the model trained on the balanced data set. The full model reaches a F-score of 0.34 for French, 0.45 for Hungarian and 0.77 for Italian. All of our approaches perform below the mean and median averages of all participants.

Surprisingly, there is a substantial difference in results obtained between the individual languages. This confirms our assumptions about the (un-) suitability of the proposed multi-lingual embedding space for this task. The results also suggest that the size of the training corpora is not influencing the final results. As seen, best results were obtained on the Italian data set were trained on the smallest corpora. Worst results were obtained on the middle, French, corpus while the biggest corpus, Hungarian, is in second place.

We identified several possible reasons for the obtained results. These also represent (possible) points for future work. One of the main disadvantages of our approach is the quality of the used word embeddings as well as the properties of the proposed language-independent embedding space. The usage of out-of-domain word embeddings which aren't targeted to the biomedical domain are likely a suboptimal solution to this problem. We tried to alleviate this by finding suitable external corpora to train domain-dependent word embeddings for each of the supported languages, however we were unable to find any significant amount of in-domain documents (e.g. PubMed search for abstracts in either French, Hungarian or Italian found 7843, 786 and 1659 articles respectively). Furthermore, we used a simple, heuristic solution by just concatenating the embeddings of all three languages to build a shared vector space.

Besides the issues with the used word embeddings, the inability to obtain full ICD-10 dictionaries for the selected languages has also negatively influenced the results. As a final limitation to our approach, lack of multi-label classification support has also been identified (i.e. not recognizing more than one death cause in a single input text).

Language	Model	Precision	Recall	F-score
French	Final-Balanced	0.494	0.246	0.329
	Final-Full	0.512	0.253	0.339
	Baseline	0.341	0.200	0.253
	Average	0.723	0.410	0.507
	Median	0.798	0.475	0.579
Hungarian	Final-Balanced	0.518	0.384	0.441
	Final-Full	0.522	0.388	0.445
	Baseline	0.243	0.174	0.202
	Average	0.827	0.783	0.803
	Median	0.922	0.897	0.910
Italian	Final-Balanced	0.857	0.685	0.761
	Final-Full	0.862	0.689	0.766
	Baseline	0.165	0.172	0.169
	Average	0.844	0.760	0.799
	Median	0.900	0.824	0.863

Table 4. Test results of the final pipeline. Final-Balanced = DCEM-Balanced + ICD-10_Extended. Final-Full = DCEM-Full + ICD-10_Extended

5 Conclusion and Future Work

In this paper we tackled the problem of information extraction of death causes in an multilingual environment. The proposed solution was focused on the setup and evaluation of an initial language-independent model which relies on a heuristic mutual word embedding space for all three languages. The proposed pipeline is divided in two steps: possible token describing the death cause are generated by using a sequence to sequence model first. Afterwards the generated token sequence is normalized to a ICD-10 code using a distinct LSTM-based classification model with attention mechanism. During evaluation our best model achieves an F-score of 0.34 for French, 0.45 for Hungarian and 0.77 for Italian. The obtained results are encouraging for further investigation however can't compete with the solutions of the other participants yet.

We detected several issues with the proposed pipeline. These issues serve as prospective future work to us. First of all the representation of the input words can be improved in several ways. The word embeddings we used are not optimized to the biomedical domain but are trained on general text. Existing work was proven that in-domain embeddings improve the quality of achieved results. Although this was our initial approach, the difficulties of finding adequate in-domain corpora for selected languages has proven to be to a hard to tackle. Moreover, the multi-language embedding space is currently heuristically defined as concatenation of the three word embeddings models for individual tokens. Creating an unified embedding space would create a truly language-independent token representation. The improvement of the input layer will be the main focus of our future work.

The ICD-10 classification step also suffers from lack of adequate training data. Unfortunately, we were unable to obtain extensive ICD-10 dictionaries for all languages and therefore can't guarantee the completeness of the ICD-10 label space. Another disadvantage of the current pipeline is the missing support for multi-label classification.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of the 6th International Conference on Learning Representations (ICLR 2018) (2018)
2. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In: Advances in Neural Information Processing Systems 28, pp. 1171–1179. Curran Associates, Inc. (2015)
3. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks **5**(2), 157–166 (1994)
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. Transactions of the Association of Computational Linguistics **5**(1), 135–146 (2017)
5. Cabot, C., Soualmia, L.F., Dahamna, B., Darmoni, S.J.: SIBM at CLEF eHealth Evaluation Lab 2016: Extracting Concepts in French Medical Texts with ECMT and CIMIND. In: CLEF 2015 Online Working Notes. CEUR-WS (2016)
6. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1724–1734. Association for Computational Linguistics, Doha, Qatar (October 2014)
7. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research **12**(Aug), 2493–2537 (2011)
8. Dermouche, M., Looten, V., Flicoteaux, R., Chevret, S., Velcin, J., Taright, N.: ECSTRA-INSERM@ CLEF eHealth2016-task 2: ICD10 Code Extraction from Death Certificates. In: CLEF 2016 Online Working Notes (2016)
9. Di Nunzio, G.M., Beghini, F., Vezzani, F., Henrot, G.: A Lexicon Based Approach to Classification of ICD10 Codes. IMS Unipd at CLEF eHealth Task. In: CLEF 2017 Online Working Notes. CEUR-WS (2017)
10. Dyer, C., Ballesteros, M., Ling, W., Matthews, A., Smith, N.A.: Transition-Based Dependency Parsing with Stack Long Short-Term Memory. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). vol. 1, pp. 334–343 (2015)
11. Ebersbach, M., Herms, R., Eibl, M.: Fusion Methods for ICD10 Code Classification of Death Certificates in Multilingual Corpora. In: CLEF 2017 Online Working Notes. CEUR-WS (2017)
12. Faruqui, M., Dyer, C.: Improving vector space word representations using multilingual correlation. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. pp. 462–471 (2014)
13. Guo, J., Che, W., Yarowsky, D., Wang, H., Liu, T.: Cross-lingual dependency parsing based on distributed representations. In: Proceedings of the 53rd Annual

Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). vol. 1, pp. 1234–1244 (2015)

14. Ho-Dac, L.M., Fabre, C., Birski, A., Boudraa, I., Bourriot, A., Cassier, M., Delvenne, L., Garcia-Gonzalez, C., Kang, E.B., Piccinini, E.: LITL at CLEF eHealth2017: automatic classification of death reports. In: CLEF 2017 Online Working Notes. CEUR-WS (2017)
15. Ho-Dac, L.M., Tanguy, L., Grauby, C., Mby, A.H., Malosse, J., Rivière, L., Veltz-Mauclair, A.: LITL at CLEF eHealth2016: recognizing entities in French biomedical documents. In: CLEF 2016 Online Working Notes. CEUR-WS (2016)
16. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. A field guide to dynamical recurrent neural networks. IEEE Press (2001)
17. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
18. Jonnagaddala, J., Hu, F.: Automatic coding of death certificates to ICD-10 terminology. In: CLEF 2017 Online Working Notes. CEUR-WS (2017)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2014)
20. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural Architectures for Named Entity Recognition. In: Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 260–270 (2016)
21. Miftahutdinov, Z., Tutubalina, E.: Kfu at clef ehealth 2017 task 1: Icd-10 coding of english death certificates with recurrent neural networks. In: CLEF 2017 Online Working Notes. CEUR-WS (2017)
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
23. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
24. van Mulligen, E.M., Afzal, Z., Akhondi, S.A., Vo, D., Kors, J.A.: Erasmus MC at CLEF eHealth 2016: Concept Recognition and Coding in French Texts. In: CLEF 2016 Online Working Notes. CEUR-WS (2016)
25. Névéol, A., Anderson, R.N., Cohen, K.B., Grouin, C., Lavergne, T., Rey, G., Robert, A., Rondet, C., Zweigenbaum, P.: CLEF eHealth 2017 Multilingual Information Extraction task overview: ICD10 coding of death certificates in English and French. In: CLEF 2017 Evaluation Labs and Workshop: Online Working Notes, CEUR-WS. p. 17 (2017)
26. Névéol, A., Cohen, K.B., Grouin, C., Hamon, T., Lavergne, T., Kelly, L., Goeuriot, L., Rey, G., Robert, A., Tannier, X., Zweigenbaum, P.: Clinical Information Extraction at the CLEF eHealth Evaluation lab 2016. CEUR workshop proceedings **1609**, 28–42 (September 2016)
27. Névéol, A., Robert, A., Grippo, F., Morgand, C., Orsi, C., Pelikán, L., Ramadier, L., Rey, G., Zweigenbaum, P.: CLEF eHealth 2018 Multilingual Information Extraction task Overview: ICD10 Coding of Death Certificates in French, Hungarian and Italian. In: CLEF 2018 Evaluation Labs and Workshop: Online Working Notes. CEUR-WS (September 2018)
28. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)

29. Peters, M., Ammar, W., Bhagavatula, C., Power, R.: Semi-supervised sequence tagging with bidirectional language models. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 1756–1765 (2017)
30. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics (2018)
31. Pham, H., Luong, T., Manning, C.: Learning distributed representations for multilingual text sequences. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing. pp. 88–94 (2015)
32. Pinter, Y., Guthrie, R., Eisenstein, J.: Mimicking Word Embeddings using Subword RNNs. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 102–112 (2017)
33. Raffel, C., Ellis, D.P.: Feed-forward networks with attention can solve some long-term memory problems. In: Workshop Extended Abstracts of the 4th International Conference on Learning Representations (2016)
34. Suominen, H., Kelly, L., Goeuriot, L., Kanoulas, E., Azzopardi, L., Spijker, R., Li, D., Névéol, A., Ramadier, L., Robert, A., Zuccon, G., Palotti, J.: Overview of the CLEF eHealth Evaluation Lab 2018. In: CLEF 2018 - 8th Conference and Labs of the Evaluation Forum. Lecture Notes in Computer Science (LNCS), Springer (2018)
35. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
36. Søgaard, A., Agić, Z., Alonso, H.M., Plank, B., Bohnet, B., Johannsen, A.: Inverted indexing for cross-lingual NLP. In: The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015) (2015)
37. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research **37**, 141–188 (2010)
38. Vyas, Y., Carpuat, M.: Sparse bilingual word representations for cross-lingual lexical entailment. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1187–1197 (2016)
39. Wang, P., Qian, Y., Soong, F.K., He, L., Zhao, H.: Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. arXiv preprint arXiv:1510.06168 (2015)
40. Wei, Q., Chen, T., Xu, R., He, Y., Gui, L.: Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. Database: The Journal of Biological Databases and Curation **2016** (2016)
41. Xing, C., Wang, D., Liu, C., Lin, Y.: Normalized word embedding and orthogonal transform for bilingual word translation. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1006–1011 (2015)