Embedding strategies for specialized domains: application to clinical entity recognition

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Pierre Zweigenbaum

▶ **To cite this version:**

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Pierre Zweigenbaum. Embedding strategies for specialized domains: application to clinical entity recognition. ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Jul 2019, Florence, Italy. pp.295-301, 10.18653/v1/P19-2041 . hal-02860947

HAL Id: hal-02860947
https://hal.science/hal-02860947v1

Submitted on 13 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Embedding Strategies for Specialized Domains: Application to Clinical Entity Recognition

Hicham El Boukkouri[1,2], Olivier Ferret[3], Thomas Lavergne[1,2], Pierre Zweigenbaum[1]

[1]LIMSI, CNRS, Université Paris-Saclay, Orsay, France,
[2]Univ. Paris-Sud,
[3]CEA, LIST, Gif-sur-Yvette, F-91191 France.
{elboukkouri,lavergne,pz}@limsi.fr, olivier.ferret@cea.fr

Abstract

Using pre-trained word embeddings in conjunction with Deep Learning models has become the *de facto* approach in Natural Language Processing (NLP). While this usually yields satisfactory results, off-the-shelf word embeddings tend to perform poorly on texts from specialized domains such as clinical reports. Moreover, training specialized word representations from scratch is often either impossible or ineffective due to the lack of large enough in-domain data. In this work, we focus on the clinical domain for which we study embedding strategies that rely on general-domain resources only. We show that by combining off-the-shelf contextual embeddings (ELMo) with static word2vec embeddings trained on a small in-domain corpus built from the task data, we manage to reach and sometimes outperform representations learned from a large corpus in the medical domain.[1]

1 Introduction

Today, the NLP community can enjoy an ever-growing list of embedding techniques that include factorization methods (e.g. GloVe (Pennington et al., 2014)), neural methods (e.g. word2vec (Mikolov et al., 2013), fastText (Bojanowski et al., 2017)) and more recently dynamic methods that take into account the context (e.g. ELMo (Peters et al., 2018), BERT (Devlin et al., 2018)).

The success of these methods can be arguably attributed to the availability of large general-domain corpora like Wikipedia, Gigaword (Graff et al., 2003) or the BooksCorpus (Zhu et al., 2015). Unfortunately, similar corpora are often unavailable for specialized domains, leaving the NLP practitioner with only two choices: either using general-domain word embeddings that are probably not fit for the task at hand or training new embeddings on the available in-domain corpus, which may probably be too small and result in poor performance.

In this paper, we focus on the clinical domain and explore several ways to improve pre-trained embeddings built from a small corpus in this domain by using different kinds of general-domain embeddings. More specifically, we make the following contributions:

- we show that word embeddings trained on a small in-domain corpus can be improved using off-the-shelf contextual embeddings (ELMo) from the general domain. We also show that this combination performs better than the contextual embeddings alone and improves upon static embeddings trained on a large in-domain corpus;
- we define two ways of combining contextual and static embeddings and conclude that the naive concatenation of vectors is consistently outperformed by the addition of the static representation directly into the internal linear combination of ELMo;
- finally, we show that ELMo models can be successfully fine-tuned on a small in-domain corpus, bringing significant improvements to strategies involving contextual embeddings.

2 Related Work

Former work by Roberts (2016) analyzed the trade-off between corpus size and similarity when training word embeddings for a clinical entity recognition task. The author's conclusion was that while embeddings trained with word2vec on in-domain texts performed generally better, a combination of both in-domain and general domain em-

[1]Python code for reproducing our experiments is available at: https://github.com/helboukkouri/acl_srw_2019

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 295–301
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

3. Echocardiogram on **DATE[Nov 6 2007] , showed ejection fraction of 55% , mild mitral insufficiency , and 1+ tricuspid insufficiency with mild pulmonary hypertension .
DERMOPLAST TOPICAL TP Q12H PRN Pain DOCUSATE SODIUM 100 MG PO BID PRN Constipation IBUPROFEN 400-600 MG PO Q6H PRN Pain
The patient had headache that was relieved only with oxycodone . A CT scan of the head showed microvascular ischemic changes . A followup MRI which also showed similar changes . This was most likely due to her multiple myeloma with hyperviscosity .

Table 1: Examples of entity mentions (**Problem**, **Treatment**, and **Test**) from the i2b2 2010 dataset*.
* This table is reproduced from (Roberts, 2016).

beddings worked the best. Subsequent work by Zhu et al. (2018) obtained state-of-the-art results on the same task using contextual embeddings (ELMo) that were pre-trained on a large in-domain corpus made of medical articles from Wikipedia and clinical notes from MIMIC-III (Johnson et al., 2016). More recently, these embeddings were outperformed by BERT representations pre-trained on MIMIC-III, proving once more the value of large in-domain corpora (Si et al., 2019).[2]

While interesting for the clinical domain, these strategies may not always be applicable to other specialized fields since large in-domain corpora like MIMIC-III will rarely be available. To deal with this issue, we explore embedding combinations[3]. In this respect, we consider both static forms of combination explored in (Yin and Schütze, 2016; Muromägi et al., 2017; Bollegala et al., 2018) and more dynamic modes of combination that can be found in (Peters et al., 2018) and (Kiela et al., 2018). In this work, we show in particular how a combination of general-domain contextual embeddings, fine-tuning, and in-domain static embeddings trained on a small corpus can be employed to reach a similar performance using resources that are available for any domain.

3 Evaluation Task: i2b2/VA 2010 Clinical Concept Detection

We evaluate our embedding strategies on the Clinical Concept Detection task of the 2010 i2b2/VA challenge (Uzuner et al., 2011).

[2]In this work, we will be focusing on contextualized embeddings from ELMo.

[3]This is more generally related to the notion of "meta-embeddings" and ensemble of embeddings as highlighted by Yin and Schütze (2016).

3.1 Data

The data consists of discharge summaries and progress reports from three different institutions: Partners Healthcare, Beth Israel Deaconess Medical Center, and the University of Pittsburgh Medical Center. These documents are labeled and split into 394 training files and 477 test files for a total of $30{,}946 + 45{,}404 \approx 76{,}000$ sequences [4].

3.2 Task and Model

The goal of the Clinical Concept Detection task is to extract three types of medical entities: problems (e.g. the name of a disease), treatments (e.g. the name of a drug) and tests (e.g. the name of a diagnostic procedure). Table 1 shows examples of entity mentions and Table 2 shows the distribution of each entity type in the training and test sets.

Entity type	Train set	Test set
Problem	11,967	18,550
Treatment	8,497	13,560
Test	7,365	12,899
Total	27,829	45,009

Table 2: Distribution of medical entity types.

To solve this task, we choose a bi-LSTM-CRF as is usual in entity recognition tasks (Lample et al., 2016; Chalapathy et al., 2016; Habibi et al., 2017). Our particular architecture uses 3 bi-LSTM layers with 256 units, a dropout rate of 0.5 and is implemented using the AllenNLP framework (Gardner et al., 2018). During training, the exact span F1 score is monitored on 5,000 randomly sampled sequences for early-stopping.

[4]Due to limitations introduced by the Institutional Review Board (IRB), only part of the original 2010 data can now be obtained for research at https://www.i2b2.org/NLP/DataSets/. Our work uses the full original dataset.

4 Embedding Strategies

We focus on two kinds of embedding algorithms: static embeddings (word2vec) and contextualized embeddings (ELMo). The first kind assigns to each token a fixed representation (hence the name "static"), is relatively fast to train but does not manage out-of-vocabulary words and polysemy. The second kind, on the other hand, produces a contextualized representation. As a result, the word embedding is adapted dynamically to the context and polysemy is managed. Moreover, in the particular case of ELMo, word embeddings are character-level, which implies that the model is able to produce vectors whether or not the word is part of the training vocabulary.

Despite contextualized embeddings usually performing better than static embeddings, they still require large amounts of data to be trained successfully. Since this data is often unavailable in specialized domains, we explore strategies that combine off-the-shelf contextualized embeddings with static embeddings trained on a small in-domain corpus.

4.1 Static Embeddings

First, we use word2vec[5] to train embeddings on a small corpus built from the task data:

i2b2 (2010) 394 documents from the training set to which we added 826 more files from a set of unlabeled documents. This is a small (1 million tokens) in-domain corpus. Similar corpora will often be available in other specialized domains as it is always possible to build a corpus from the training documents.

Then, we also train embeddings on each of two general-domain corpora:

Wikipedia (2017) encyclopedia articles from the 01/10/2017 data dump[6]. This is a large (2 billion tokens) corpus from the general domain that has limited coverage of the medical field.

Gigaword (2003) newswire text data from many sources including the New York Times. This is a large (2 billion tokens) corpus from the general domain with almost no coverage of the medical field.

[5]We used the following parameters: `cbow=1`, `size=256`, `window=5`, `min-count=5`, `iter=10`.

[6]Similar dumps can be downloaded at https://dumps.wikimedia.org/enwiki/.

4.2 Contextualized Embeddings

We use two off-the-shelf ELMo models[7]:

ELMo_small a general-domain model trained on the 1 Billion Word Benchmark corpus (Chelba et al., 2013). This is the small version of ELMo that produces 256-dimensional embeddings.

ELMo_original the original ELMo model. This is a general-domain model trained on a mix of Wikipedia and newswire data. It produces 1024-dimensional embeddings.

Additionally, we also build embeddings by fine-tuning each model on the i2b2 corpus. The fine-tuning is achieved by resuming the training of the ELMo language model on the new data (i2b2). At each epoch, the validation perplexity is monitored and ultimately the best model is chosen:

ELMo_small$_{finetuned}$ the result of fine-tuning ELMo_small for 10 epochs.

ELMo_original$_{finetuned}$ the result of fine-tuning ELMo_original for 5 epochs.

4.3 Embedding Combinations

There are many possible ways to combine embeddings. In this work, we explore two methods:

Concatenation a simple concatenation of vectors coming from two different embeddings. This is denoted $\mathbf{X}\oplus\mathbf{Y}$ (e.g. i2b2$\oplus$Wikipedia).

Mixture in the particular case where ELMo embeddings were combined with word2vec vectors, we can directly add the word2vec embedding in the linear combination of ELMo. We denote this combination strategy $\mathbf{X}+\mathbf{Y}$ (e.g. ELMo_small+i2b2).

The mixture method generalizes the way ELMo representations are combined. Given a word w, if we denote the three internal representations produced by ELMo (i.e. the CharCNN, 1^{st} bi-LSTM and 2^{nd} bi-LSTM representations) by h_1, h_2, h_3, we recall that the model computes the word's embedding as:

$$\text{ELMo}(w) = \gamma(\alpha_1 h_1 + \alpha_2 h_2 + \alpha_3 h_3)$$

[7]All the models with their descriptions are available at https://allennlp.org/elmo.

Embedding Strategy	X	i2b2 $\oplus$ X	i2b2 $+\!\!+$ X
i2b2	82.06 $\pm$ 0.32	-	-
Wikipedia	83.30 $\pm$ 0.25	83.35 $\pm$ 0.62	-
Gigaword	82.54 $\pm$ 0.41	83.10 $\pm$ 0.37	-
ELMo_small	80.79 $\pm$ 0.95	84.18 $\pm$ 0.26	84.94 $\pm$ 0.94
ELMo_original	84.28 $\pm$ 0.66	85.25 $\pm$ 0.21	85.64 $\pm$ 0.33
ELMo_small$_{finetuned}$	83.86 $\pm$ 0.87	84.81 $\pm$ 0.40	85.93 $\pm$ 1.01
ELMo_original$_{finetuned}$	85.90 $\pm$ 0.50	**86.18** $\pm$ 0.48	**86.23** $\pm$ 0.58

Table 3: Performance of various strategies involving a general-domain resource and a small in-domain corpus (i2b2). The values are Exact Span F1 scores given as Mean $\pm$ Std (bold: best result for each kind of combination).

where γ and $\{\alpha_i, i = 1, 2, 3\}$ are tunable task-specific coefficients[8]. Given h_{w2v}, the word2vec representation of the word w, we compute a "mixture" representation as:

$$\text{ELMo}_{\text{mix}}(w) = \gamma(\alpha_1 h_1 + \alpha_2 h_2 + \alpha_3 h_3 + \beta h_{w2v})$$

where β is a new tunable coefficient[9].

5 Results and Discussion

We run each experiment with 10 different random seeds and report performance in mean and standard deviation (std). Values are expressed in terms of strict F1 measure that we compute using the official script from the i2b2/VA 2010 challenge.

5.1 Using General-domain Resources

Table 3 shows the results we obtain using general-domain resources only. The top part of the table shows the performance of word2vec embeddings trained on i2b2 as well as two general-domain corpora: Wikipedia and Gigaword. We see that i2b2 performs the worst despite being trained on in-domain data. This explicitly showcases the challenge faced by specialized domains and confirms that training embeddings on small in-domain corpora tends to perform poorly. As for the general domain embeddings, we can observe that Wikipedia is slightly better than Gigaword. This can be explained by the fact that the former has some medical-related articles which implies a better coverage of the clinical vocabulary compared to the newswire corpus Gigaword[10]. We can also see that combining general-domain word2vec embeddings with i2b2 results in weak improvements that are slightly higher for Gigaword probably for the same reason.

The middle part of the table shows the results we obtain using off-the-shelf contextualized representations. Looking at the embeddings alone, we see that ELMo_small performs worse than i2b2 while ELMo_original is better than all word2vec embeddings. Again, the reason for the small model's performance might be related to the different training corpora. In fact, ELMo_original, aside from being a larger model, was trained on Wikipedia articles which may include some medical articles. Another interesting point is that both the mean and variance of the performance when using off-the-shelf ELMo models improve notably when combined with word2vec embeddings trained on i2b2. This improvement is even greater for the small model, probably because it has less coverage of the medical domain. Furthermore, we see that the performance improves again, although to a lesser extent when the word2vec embedding is mixed with ELMo instead of combined through concatenation.

The bottom part of the table shows the results obtained after fine-tuning both ELMo models. We see that fine-tuning improves all the results (but to varying extents), with the best performance being achieved using combinations—either concatenation or mixture—of i2b2's word2vec and the larger fine-tuned ELMo.

Two points are worth being noted. First, it is interesting to see that we achieve good results with a model that only uses an off-the-shelf model and a small in-domain corpus built from the task data. This is a valuable insight since the same strategy could be applied for any specialized

[8] In practice, the coefficients go through a softmax before being used in the linear combination.

[9] In particular cases where the ELMo model produces 1024-dimensional embeddings, we duplicate the 256-dimensional word2vec embeddings so that the dimensions match before mixing.

[10] We count 14.42% out-of-vocabulary tokens in Gigaword against 5.82% for Wikipedia.

domain. Second, we see that the smaller 256-dimensional ELMo model, which initially performed very poorly (≈ 80 F1), improved drastically (≈ +6 F1) using our best strategy and does not lag very far behind the original 1024-dimensional model. This is also valuable since many practitioners do not have the computational resources that are required for using the larger versions of recent models like ELMo or BERT.

5.2 Using In-domain Resources

It is natural to wonder how our results fare against models trained on large in-domain corpora. Fortunately, there are two such corpora in the clinical domain:

MIMIC III (2016) a collection of medical notes from a large database of Intensive Care Unit encounters at a large hospital (Johnson et al., 2016)[11]. This is a large (1 billion tokens) in-domain corpus.

PubMed (2018) a collection of scientific article abstracts in the biomedical domain[12]. This is a large (4 billion tokens) corpus from a close but somewhat different domain.

Both Zhu et al. (2018) and Si et al. (2019) trained the ELMo (original) on MIMIC, with the former resorting to only a part of MIMIC mixed with some curated Wikipedia articles. Table 4 reports their results, to which we add the performance of strategies using word2vec embeddings trained on MIMIC and PubMed, and an open-source ELMo model trained on PubMed[13].

We can see yet again that word2vec embeddings perform less well than ELMo models trained on the same corpora. We also see that combining the two kinds of embeddings still brings some improvement (see ELMo (PubMed) + MIMIC). And more importantly, we observe that by using only general-domain resources, we perform very close to the ELMo models trained on a large in-domain corpus (MIMIC) with a maximum difference in F1 measure of ≈ 1.5 points.

[11]The MIMIC-III corpus can be downloaded at https://mimic.physionet.org/gettingstarted/access/.

[12] The PubMed-MEDLINE corpus can be downloaded at https://www.nlm.nih.gov/databases/download/pubmed_medline.html.

[13]Since we did not train this model ourselves, we are not sure whether the training corpus is equivalent to the PubMed corpus we use for training word2vec embeddings.

Embedding Strategy	F1
MIMIC	84.29 ± 0.30
PubMed	84.06 ± 0.14
ELMo (PubMed)	86.29 ± 0.61
ELMo (PubMed) + MIMIC	87.17 ± 0.54
ELMo_original$_{finetuned}$ + i2b2	86.23 ± 0.58
ELMo (Clinical) (Zhu et al., 2018)	86.84 ± 0.16
ELMo (MIMIC) (Si et al., 2019)	**87.80**

Table 4: Comparison of strategies using large in-domain corpora with the best strategy using a small in-domain corpus and general-domain resources. The values are Exact Span F1 scores.

5.3 Using GloVe and fastText

In order to make sure that the observed phenomena are not the result of using the word2vec method in particular, we reproduce the same experiments using GloVe and fastText[14]. The corresponding results are reported in Table 5 and Table 6.

We can see that GloVe and fastText are always outperformed by word2vec when trained on a single corpus only. This is not true anymore when combining these embeddings with representations from ELMo. In fact, in this case, the results are mostly comparable to the performance obtained when using word2vec, with a slight improvement when using fastText. This small improvement may be explained by the fact that the fastText method is able to manage Out-Of-Vocabulary tokens while GloVe and word2vec are not.

More importantly, these additional experiments validate the initial results obtained with word2vec: static embeddings pre-trained on a small in-domain corpus (i2b2) can be combined with general domain contextual embeddings (ELMo), through either one of the proposed methods, to reach a performance that is comparable to the state-of-the-art[15].

5.4 Limitations

We can list the following limitations for this work:

- we tested only one specialized domain on one task using one NER architecture. Although

[14]We used the following parameters: (GloVe) `size=256, window=15, min-count=5, iter=10`; (fastText) `skipgram, size=256, window=5, min-count=5, neg=5, loss=ns, minn=3, maxn=6, iter=10`.

[15]Our single best model gets a F1 score of 87.10.

Embedding Strategy	X	i2b2 $\oplus$ X	i2b2 $+\!\!+$ X
i2b2	80.21 $\pm$ 0.37	-	-
Wikipedia	81.82 $\pm$ 0.52	81.29 $\pm$ 0.42	-
Gigaword	81.38 $\pm$ 0.33	81.47 $\pm$ 0.18	-
ELMo_small	80.79 $\pm$ 0.95	83.04 $\pm$ 1.03	84.30 $\pm$ 0.72
ELMo_original	84.28 $\pm$ 0.66	85.00 $\pm$ 0.32	85.12 $\pm$ 0.26
ELMo_small$_{finetuned}$	83.86 $\pm$ 0.87	84.42 $\pm$ 0.75	85.19 $\pm$ 0.75
ELMo_original$_{finetuned}$	85.90 $\pm$ 0.50	86.05 $\pm$ 0.16	**86.46** $\pm$ 0.36

Table 5: Performance of the strategies from Table 3 using GloVe instead of word2vec (bold: GloVe > word2vec)

Embedding Strategy	X	i2b2 $\oplus$ X	i2b2 $+\!\!+$ X
i2b2	81.98 $\pm$ 0.41	-	-
Wikipedia	82.32 $\pm$ 0.37	81.84 $\pm$ 1.48	-
Gigaword	81.77 $\pm$ 0.36	82.40 $\pm$ 0.32	-
ELMo_small	80.79 $\pm$ 0.95	**84.44** $\pm$ 0.42	**85.47** $\pm$ 0.61
ELMo_original	84.28 $\pm$ 0.66	**85.57** $\pm$ 0.46	**85.77** $\pm$ 0.47
ELMo_small$_{finetuned}$	83.86 $\pm$ 0.87	**85.18** $\pm$ 0.67	**86.27** $\pm$ 0.35
ELMo_original$_{finetuned}$	85.90 $\pm$ 0.50	**86.49** $\pm$ 0.28	**86.82** $\pm$ 0.29

Table 6: Performance of the strategies from Table 3 using fastText instead of word2vec (bold: fastText > word2vec)

the results look promising, they should be validated by a wider set of experiments;

- our best strategies use the task corpus (i2b2) to adapt general off-the-shelf embeddings to the target domain, then combine two different types of embeddings as an ensemble to boost performance. This may not work if the task corpus is really small (we recall that our corpus is $\approx$ 1 million tokens).

6 Conclusion and Future Work

While embedding methods are improving on a regular basis, specialized domains still lack large enough corpora to train these embeddings successfully. We address this issue and propose embedding strategies that only require general-domain resources and a small in-domain corpus. In particular, we show that using a combination of general-domain ELMo, fine-tuning and word2vec embeddings trained on a small in-domain corpus, we achieve a performance that is not very far behind that of models trained on large in-domain corpora. Future work may investigate other contextualized representations such as BERT, which has proven to be superior to ELMo—at least on our task—in the recent work by Si et al. (2019). Another interesting research direction could be exploiting external knowledge (e.g. ontologies) that may be easier to find in specialized fields than large corpora.

Acknowledgments

This work has been funded by the French National Research Agency (ANR) and is under the ADDICTE project (ANR-17-CE23-0001). We are also thankful to Kirk Roberts for having kindly provided assistance regarding his 2016 paper.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Danushka Bollegala, Koheu Hayashi, and Ken ichi Kawarabayashi. 2018. Think globally, embed locally — locally linear meta-embedding of words. In *Proceedings of IJCAI-ECAI*, pages 3970–3976.

Raghavendra Chalapathy, Ehsan Zare Borzeshi, and Massimo Piccardi. 2016. Bidirectional LSTM-CRF for clinical concept extraction. In *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, pages 7–12, Osaka, Japan. The COLING 2016 Organizing Committee.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. 2017. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic meta-embeddings for improved sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 1466–1477, Brussels, Belgium.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR 20013), workshop track*.

Avo Muromägi, Kairit Sirts, and Sven Laur. 2017. Linear ensembles of word embedding models. In *21st Nordic Conference on Computational Linguistics (NoDaLiDa 2017)*, pages 96–104, Gothenburg, Sweden.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Kirk Roberts. 2016. Assessing the corpus size vs. similarity trade-off for word embeddings in clinical NLP. In *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, pages 54–63.

Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. 2019. Enhancing clinical concept extraction with contextual embedding. *arXiv preprint arXiv:1902.08691*.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1351–1360, Berlin, Germany.

Henghui Zhu, Ioannis Ch. Paschalidis, and Amir Tahmasebi. 2018. Clinical concept extraction with contextual word embedding. In *Proceedings of the Machine Learning for Health (ML4H) Workshop at NeurIPS 2018*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.