Word-Level Loss Extensions for Neural Temporal Relation Classification

Artuur Leeuwenberg and **Marie-Francine Moens**
Department of Computer Science
KU Leuven, Belgium
{*tuur.leeuwenberg, sien.moens*}@*cs.kuleuven.be*

Abstract

Unsupervised pre-trained word embeddings are used effectively for many tasks in natural language processing to leverage unlabeled textual data. Often these embeddings are either used as initializations or as fixed word representations for task-specific classification models. In this work, we extend our classification model's task loss with an unsupervised auxiliary loss on the word-embedding level of the model. This is to ensure that the learned word representations contain both task-specific features, learned from the supervised loss component, and more general features learned from the unsupervised loss component. We evaluate our approach on the task of temporal relation extraction, in particular, narrative containment relation extraction from clinical records, and show that continued training of the embeddings on the unsupervised objective together with the task objective gives better task-specific embeddings, and results in an improvement over the state of the art on the THYME dataset, using only a general-domain part-of-speech tagger as linguistic resource.

1 Introduction

Word representations in the form of continuous vectors are often pre-trained on large amounts of raw text to learn general word features, using unsupervised objectives. These representations are then used in supervised models for various classification tasks. However, such tasks sometimes require very specific features that may not have been captured by the unsupervised objective. In other domains such as computer vision, representations are often learned jointly from multiple resources for classification. In this work, we explore the possibility to exploit learning signals from both settings to construct better task-oriented word representations, and obtain a better relation classification model.

The main task in this work is the extraction of narrative containment relations (CR) from English clinical texts, as annotation of clinical data is costly and it is therefor crucial to fully exploit both the labeled as well as the unlabeled data that is available. The aim of CR extraction is to find if, given events A and B, event A is temporally contained in event B (i.e. if event A happens within the time span of event B). An example of such relation is given in Fig. 1, where the model should predict all containment edges given the entities (events and temporal expressions), by classifying each pair of entities as containment or no containment. Temporal relation classification in clinical text is a very important task in the secondary use of clinical data from electronic health records. The patient time-line is crucial for making a good patient prognosis and clinical decision support (Onisko et al., 2015). This task has already been addressed

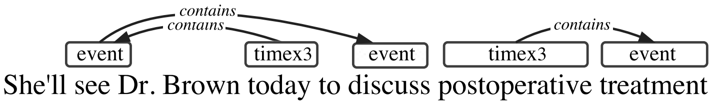
contains
contains
contains
event
timex3
event
timex3
event
She'll see Dr. Brown today to discuss postoperative treatment

Figure 1: A sentence annotated with events, temporal expressions (Timex3), and containment relations.

Proceedings of the 27th International Conference on Computational Linguistics, pages 3436–3447
Santa Fe, New Mexico, USA, August 20-26, 2018.

in three iterations of the Clinical TempEval Shared Task (Bethard et al., 2016). Still there is a gap of more than 0.20 in F-measure between the state-of-the-art CR extraction systems and the inter-adjunctator agreement (indicating an upper bound for performance). This shows that this task is very challenging.

Following the current trend in NLP, the recent state-of-the-art models for extraction of CR are neural network models. These models all use pre-trained word embeddings as word representations (Tourille et al., 2017; Dligach et al., 2017; Lin et al., 2017).

Pre-training of the embeddings is done with an auxiliary task (a task where one is not interested in the final predictions, but in the trained model components), like the skip-gram task (Mikolov et al., 2013). When used for classification tasks in NLP, these pre-trained word representations are often either used as fixed inputs for the classification model, or as initialization for the word representations of the classification model (sometimes called fine-tuned embeddings).

A problem with pre-trained representations in classification models is that solving the main task often requires different information than the auxiliary task. Training word representations only on the auxiliary objective can result in loss of crucial information for the task, and afterwards fine-tuning on the task loss does not influence words that are not in the task's training data, losing generalization to those words.

In the current work, we propose a neural relation classification (RC) model that learns its word representations jointly on the main task (supervised, on labeled data) and on the auxiliary task (unsupervised, on unlabeled data) in a multi-task setting to overcome this problem, and ensure that the embeddings contain valuable information for our main task, while still leveraging the unlabeled data for more general feature learning. As auxiliary task we implement a skip-gram (SG) architecture, similar to Mikolov et al. (2013). Our proposed models use only unlabeled data and a general (news, out of domain) part-of-speech (POS) tagger as external resources, in contrast to the current state-of-the-art models, to ease extension to other languages for which specialized NLP tools for clinical texts might not be available. The main contributions of this work are that it:

- Shows that training the word-level representations jointly on its main task and an auxiliary objective results in better representations for classification, compared to using pre-trained variants.

- Shows that the method's increased performance and hyper-parameters are robust across different training set sizes, and that single-loss training settings act as lower bounds on performance.

- Constitutes a new state of the art for temporal relation extraction on the THYME dataset even without dedicated clinical preprocessing.

2 Related Work

The model we present draws inspiration from prior research on (temporal) relation classification and neural multi-task learning.

2.1 Clinical Temporal Relation Extraction

Temporal relation extraction from clinical texts is a widely studied area in NLP and has been explored through various shared tasks, such as the i2b2 shared task on clinical temporal information (Sun et al., 2013), and three iterations of Clinical TempEval (Bethard et al., 2015; Bethard et al., 2016; Bethard et al., 2017). Until recently, most of the top performing systems employed manually constructed linguistic feature sets (Lin et al., 2015; Lee et al., 2016; Leeuwenberg and Moens, 2017). In the last few years, there has been a shift towards using neural models, using LSTM (Tourille et al., 2017) and CNN models (Dligach et al., 2017; Lin et al., 2017) inspired by the work on relation classification in other domains (Zeng et al., 2014; Zhang and Wang, 2015; Zhou et al., 2016; Nguyen and Grishman, 2015). The top results in clinical temporal relation extraction are still achieved when enhancing the neural models with dedicated clinical NLP tools for preprocessing the clinical texts, often using the English cTAKES system (Savova et al., 2010), which contains tools for clinical POS tagging, named entity recognition, and a dependency parser all trained on clinical data. The main reason for using these dedicated clinical tools is that parsers trained on non-clinical texts perform significantly worse on clinical data (Jiang et al., 2015).

Dedicated clinical NLP tools are not available for most languages though, and retraining NLP tools on clinical data is quite resource intensive, because it requires extra annotation effort. Additionally, clinical data is often difficult to obtain or share publicly for patient privacy reasons. Hence, we keep resource intensive preprocessing to a minimum and employ only a general news domain POS tagger (Toutanova et al., 2003), providing important temporal relation extraction cues, such as tense shifts (Derczynski, 2017), and for which training data are available for many languages (Petrov et al., 2012).

2.2 Multi-task Learning

Our proposed model training can be seen as multi-task learning (MTL), where the aim is to improve model generalization by leveraging the information from training signals of different related tasks (Caruana, 1998). In earlier work, MTL has shown to be quite effective for different NLP tasks such as machine translation (Dong et al., 2015), sentiment analysis (Peng and Dredze, 2015; Yu and Jiang, 2016), sentence level name prediction (Cheng et al., 2015), semantic role labeling (Collobert and Weston, 2008), and many more. For example, Collobert and Weston (2008) used an auxiliary unsupervised objective for semantic role labeling (SRL). They alternately trained embeddings in a language model and a SRL model. In contrast to their work, we learn both tasks truly jointly, and optimize a single semi-supervised objective. Typically in neural MTL, one or more layers of the network are shared among different models. Two issues in MTL are (1) how to determine if the tasks are related enough to benefit from each other, and (2) what layers to share among the models. Baxter and others (2000) theoretically argue that tasks are related when they share an inductive bias. In our model, we expect that the skip-gram task (Mikolov et al., 2013) can act as a reasonable word-level inductive bias for our task, as it has already shown its effectiveness in SRL (Collobert and Weston, 2008) and sentiment analysis (Peng and Dredze, 2015) in MTL, and for many NLP classification tasks when using them as pre-trained embeddings. Hashimoto et al. (2017) showed that even when combining many tasks, considering the task hierarchy (simpler tasks lower in the network) allows them to benefit from each other. In most work on MTL the auxiliary tasks are supervised and specifically chosen for their relatedness to the main task (Ruder, 2017), whereas in our model we chose the unsupervised auxiliary skip-gram task, and share weights of the word embedding layer. This results in a new joint relation classification objective that is semi-supervised on the word-level and provides better generalization for the final classification model.

3 The Model

Our model consist of two components: (1) a relation classification component (RC), and (2) a skip-gram component (SG). A high-level schematic overview of our model's training setting is shown in Fig. 2.

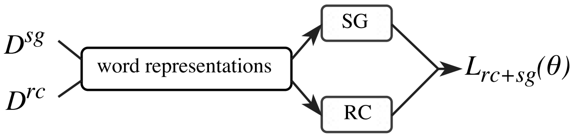

Figure 2: High-level overview of our model training setting. D^{rc} and D^{rc} indicate the dataset for the relation classification and skip-gram components respectively, and $L_{rc+sg}(\theta)$ the model's combined loss.

3.1 Relation Classification (RC)

To classify relations we employ a long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) relation classification model (RC) (Zhang and Wang, 2015). We frame the task as a sequence classification problem, taking as an input: the textual candidate relation description, i.e. the arguments of the candidate relation (the entity pair), and the context words surrounding the arguments, all read as a sequence from left to right. A schematic overview of the RC model component is shown in Fig. 3.

Generation of candidate entity pairs is described later on in section 4.4. The locations of the arguments of each candidate relation are indicated by two types of features taken from the literature: (1) position indicators, which are XML tags added to the original input sequence indicating the start and end of the

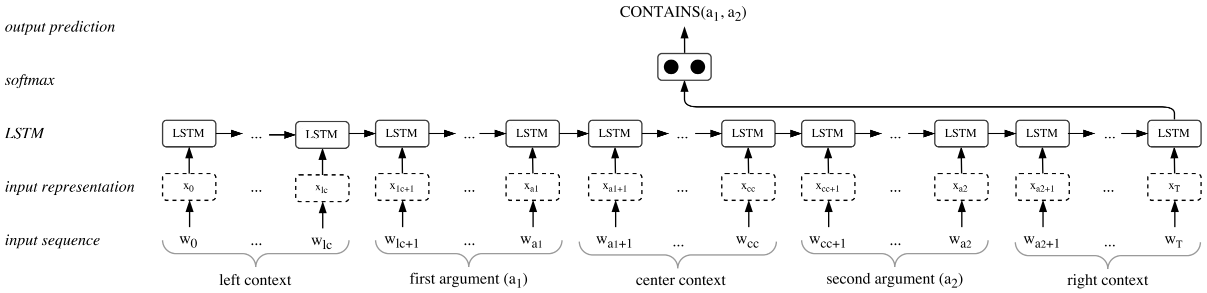

Figure 3: Schematic representation of the relation classification (RC) model component. Arrows represent sets of fully connected weights. The dashed box indicates a word input as shown in Fig. 4.

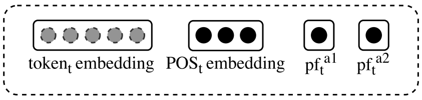

Figure 4: Each word input x_t of the RC model (at time step t) is a concatenation of a token embedding, a POS embedding, and two positional features (one for each argument).

arguments (Zhang and Wang, 2015), and (2) by position features, indicating the relative token-distance of each word to each argument (Zeng et al., 2014). Each word's total input x_t at time step $t \in \langle 0, 1, ..., T \rangle$ consists of the two argument locating features, pf_t^{a1} and pf_t^{a2}, together with a word embedding $x_t^{token} \cdot W_{em}^{token}$, and a POS embedding $x_t^{pos} \cdot W_{em}^{pos}$, where W_{em}^{token} and W_{em}^{pos} are the embedding matrices for tokens and POS respectively, and x_t^{token} and x_t^{pos} their one-hot representations. A schematic overview of the concatenated input x_t for each word to the LSTM unit is shown in Fig. 4.

The predicted class probabilities $\hat{p}_{rc}(x)$ are given by a softmax classifier placed on top of the LSTM output h at the last time step T (in Eq. 1).[1]

$$\hat{p}_{rc}(x) = \mathit{softmax}(W_p h_T + b_p) \tag{1}$$

The RC model's loss function is cross-entropy loss, as shown in Eq. 2. D^{rc} indicates the supervised relation classification dataset, and θ^{rc} is the collection of all trainable parameters of the model.

$$L_{rc}(\theta^{rc}) = - \sum_{i=1}^{|D^{rc}|} y_i \log \hat{p}_{rc}(x_i) \tag{2}$$

3.2 Context Prediction (SG)

As the unsupervised auxiliary task, we implemented a feed-forward neural network for a word context prediction task, known as the continuous skip-gram (SG), following Mikolov et al. (2013). As input, the model takes a one-hot encoded input word w_j, which is projected to a word embedding, from which the probability distribution y over its surrounding context words $w_{j-c}, ..., w_{j-1}, w_{j+1}, ..., w_{j+c}$ is predicted, given a context window size c. The full model is given by Eq. 3.

$$\hat{p}_{sg}(w_j) = softmax(W_{p_{sg}}(w_j \cdot W_{em}^{token}) + b_{p_{sg}}) \tag{3}$$

Like the RC model, we use cross-entropy loss for our SG model, as shown in Eq. 4. D^{sg} indicates the unsupervised dataset, consisting of words and their contexts. θ^{sg} is the collection of all trainable parameters of our model.

$$L_{sg}(\theta^{sg}) = - \sum_{i=1}^{|D^{sg}|} y_i \log \hat{p}_{sg}(w_i) \tag{4}$$

[1] We also experimented with bidirectional LSTMs (Zhang et al., 2015) and adding attention (Zhou et al., 2016). In our experiments, this did not result in significant improvements.

3.2.1 Separate Left & Right Context (SGLR)

The skip-gram model is quite rough in its context description and does not take into account word order very well. However, for temporal relations we expect word order to be relevant. For this reason, we also experimented with a variation on the skip-gram model, separating the left and right context, following the intuition of Ling et al. (2015). The context separation is achieved by extending the context words by a 'left' or 'right' prefix depending on their location relative to the sampled word.

3.3 Combination (RC + SG)

We train our proposed model on a combination of both loss functions, each with their own dataset D^{rc}, and D^{sg} respectively. The combined loss, shown in Eq. 5, is a weighted sum of their cross-entropy losses, where λ_{sg} determines the importance of the SG loss.

$$L_{rc+sg}(\theta) = L_{rc}(\theta^{rc}) + \lambda_{sg} L_{sg}(\theta^{sg}) \tag{5}$$

A crucial part of our model is that although both models sample different types of inputs (the RC: sequences, the SG: single words) from different datasets, and have different classification weights, the word embeddings are shared, i.e. W_{em}^{token} ($\theta^{rc} \cap \theta^{sg} = W_{em}^{token}$), also illustrated in Fig. 2. So only the word embeddings are directly influenced by both losses. All other weights (from RC or SG) are only influenced indirectly, through the word embedding weights, as both models are trained simultaneously. Søgaard and Goldberg (2016) showed that, for NLP, sharing representations at the lower levels of the network is most effective: when lower level features are shared, there is room for the model to learn task specific abstractions in higher layers. For this reason we choose our model to share only the word embedding layer, as schematically illustrated in Fig. 5.

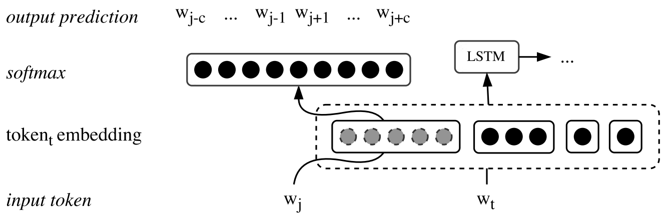

Figure 5: Schematic representation of how the SG model component extends the RC model when using the combined loss on input word $w_j \in D^{sg}$, and word w_t at time step t from input sequence $x_i \in D^{rc}$. The gray layer indicates the shared word embedding parameters. The dashed box represents the total word input x_t for RC, as in Fig. 3 and 4.

3.4 Training

We train all our models for at least 10 epochs, using Adam (Kingma and Ba, 2014) stochastic gradient descent, with the default parameters from the original paper (lr=0.001), and a batch size of 1024 on a Titan X GPU. As stopping criteria we employ early stopping (Morgan and Bourlard, 1990) with a patience of 20 epochs, based on F-measure on a small validation set of 3 documents (from Train). After each epoch on the main task, we shuffle the data and start the next one. During training we employ a dropout of 0.5 on the input, and on the second last layer (Srivastava et al., 2014).

For our semi-supervised setting, with the combined loss, we sample 1024 samples in our batch for each task from the corresponding dataset, and do a single weight update on the combined loss. A high-level schematic overview of our semi-supervised training setting is shown in Fig. 2.

4 Experimental Setup

4.1 Datasets

We conduct our experiments on the THYME corpus (Styler IV et al., 2014), a temporally annotated corpus of clinical notes in the colon cancer domain, also used in the Clinical TempEval Shared Tasks

(Bethard et al., 2015; Bethard et al., 2016). We use the provided Train, Dev and Test split so we can directly compare to other approaches from the literature. The Train section consists of 195 documents, with 11,2k annotated candidate relations, the Dev section of 98 documents and 6,2 annotated candidates, and the Test section contains 100 documents and 5,9k candidates. We now refer to this dataset as D^{rc}.

As data for the SG(LR) loss, we use the raw THYME train texts, extended with a MIMIC III (Johnson et al., 2016) section of 500 discharge summaries that contain the terms 'colon' and 'cancer' at least twice. We refer to this dataset as D^{sg} and it is used in all pre-training and joint training settings.

4.2 Training Settings

We compare five training settings for the model of which the first three settings act as baselines to compare with our proposed models (settings 4 and 5):

1. **RC (random initialization):** Uses random word embedding initializations (picked from [0.05, 0.05]) and trains on loss L_{rc}.
2. **RC (SG initialization):** Initializes the model with pre-trained SG embeddings, and trains on L_{rc}.
3. **RC (SG fixed):** Initializes the model with pre-trained SG embeddings, and trains the model on L_{rc}, while not updating the word embedding weights, keeping them as fixed features.
4. **RC + SG:** Initializes the model with pre-trained SG embeddings, and trains on L_{rc+sg}.
5. **RC + SGLR:** Initializes the model with pre-trained SG embeddings, and trains on $L_{rc+sglr}$.

4.3 Evaluation

As evaluation metrics we use precision, recall, and F-measure, calculated using the evaluation script provided by the Clinical TempEval organizers[2], which evaluates the CR under the temporal closure (UzZaman et al., 2012), taking into account transitivity properties of the temporal relations.

4.4 Preprocessing & Hyper-parameters

As preprocessing of the corpus, we employ very simple tokenization: splitting the text on spaces and considering punctuation[3] and newlines as individual tokens. Additionally, we lowercase the corpus, and conflate digits (1992 → 5555). To extract POS we use the Stanford POS Tagger v3.7 (Toutanova et al., 2003), using the pre-trained (on WSJ) caseless left-3-words model. Finally, all 1-time occurring tokens in the training dataset are replaced by a <UNK>-token, to represent out-of-vocabulary words at test-time.

We employ the same candidate generation as Leeuwenberg and Moens (2017), considering all pairs of events (Event×Event, or EE) and events and temporal expressions (Timex3×Event, or TE) with a maximum token distance of 30 as candidate relations to be classified (ignoring sentence boundaries, as relations also occur across them). This candidate generation has a maximum recall of 0.87%, and gives a ratio between the positive and negative class of 1:36, also indicating the task's difficulty.

In our experiments, we tuned each model type within the same hyper-parameter search space on the Dev set. The number of LSTM units was chosen from $\{25, 50, 100\}$, and the word embedding dimension from $\{25, 50, 100\}$. This resulted in 100 LSTM units, and a word embedding size of 25. The loss weights λ_{sg} and λ_{sglr} were chosen from $\{0.01, 0.1, 1.0, 10, 100\}$, resulting in λ_{sg}=0.1 and λ_{sglr}=0.1. The context window size of the skip-gram was set to 2, chosen from $\{2, 4, 8\}$. The context size for the RC, and POS embedding dimension size were not tuned and set to 10 (left and right), and 40 respectively.

5 Results

5.1 Influence of Word-Level Loss

We looked at model performance when increasing the importance of the auxiliary word-level loss (λ_{sg} and λ_{sglr}). The results when changing these hyper-parameters are shown in Fig. 6.

[2] https://github.com/bethard/anaforatools

[3] ,./\"'=+-;:()!?<>%&$*|[]{}

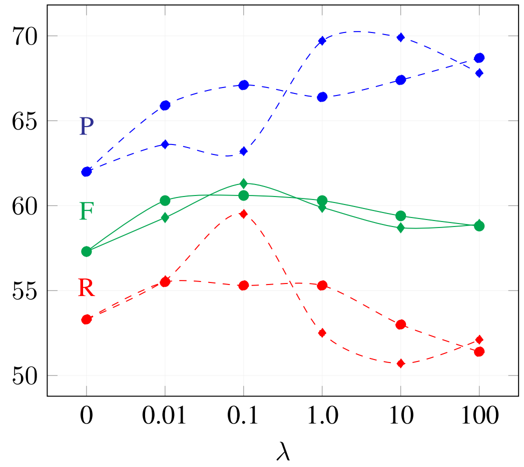

Figure 6: Precision (P), Recall (R) and F-measure (F) on the THYME Dev set for different values of λ_{sg} ($\circ$) and λ_{sglr} ($\diamond$).

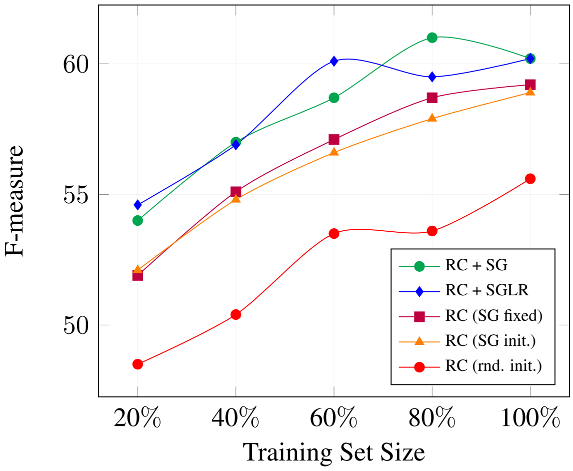

Figure 7: F-measure on the THYME Dev set for different training settings, over different training set sizes (in % of the full train set).

When choosing $\lambda = 0$ (λ_{sg} or λ_{sglr}) for our model, we obtain the same model as RC (SG init.), as the auxiliary objective has no influence. For very high values of λ, we hypothesize that the models converge towards RC (SG fixed), because when taking $\lambda \rightarrow \infty$, the word embeddings are solely optimized for the auxiliary loss, as the influence of the task loss is proportionally zero, i.e. $\lim_{\lambda \rightarrow \infty} \frac{1}{\lambda} = 0$. This property is interesting, as it shows these baseline models can act as lower bounds for our model performance when choosing a bad λ value. This can be observed in Fig. 6, where the F-measure is highest for a λ that balance both objectives, whereas for extremes F-measure decreases.

5.2 Comparison Across Training Set Size

We evaluated all model settings for different training set sizes. From Fig.7 we can see that the worst model is the one with random word embedding initializations. One improvement is to initialize the model with pre-trained SG embeddings. Fixing the pre-trained embeddings or continued training on the main task objective does not result in very different F-measure scores. However, continued training on the combined objective does seem to give a significant increase in F-measure, consistent over different training sizes for both the RC + SG as well as the RC + SGLR variant. Additionally, it should be noticed that parameters are not returned on each dataset size, but obtained from tuning on the full Dev set. Still the model ranking is consistent.

5.3 Evaluation on Subsets of Relations

To get a more detailed insight in what each model learns relative to the others, we evaluated our models on different subsets of the data. First, we split the containment relations based on their argument types and separately evaluated the 3.3k EE relations and the 2.7k TE relations. EE relations are generally found more difficult than TE relations (Lin et al., 2015; Lin et al., 2016; Dligach et al., 2017; Lin et al., 2017). In Table 1 we can see that also for our model, EE relations are harder to recognize than the TE relations, as all models achieve higher scores for TE compared to EE relations. What is interesting to see is that when training with the combined loss (SG or SGLR) we obtain a clear improvement on the more difficult EE relations, and perform slightly worse on TE relations compared to using pre-trained embeddings (the three upper settings). The reason could be that EE relations are more diverse in vocabulary, and are consequently more influenced by the quality of the embeddings.

We also analyzed the models w.r.t. total frequency in the training data ($D^{rc} + D^{sg}$) and made three subsets based on the average word frequency of the argument tokens in each relation. The three buckets of relations, 0-100, 100-500, and 500+, are of sizes 2.2k, 2.2k, and 1.8k respectively. What can be observed is that the RC+SG model performs best for low-frequency words, and RC+SGLR performs

Table 1: Evaluation on subsets of THYME Dev (in F-measure). The subsets of Event×Event (EE) and Timex3×Event (TE) relation pairs are of sizes 3.3k and 2.7k respectively. The intervals 0-100, 100-500 and 500+ are subsets reflecting average argument token frequency in the training data (of sizes 2.2k, 2.2k and 1.8k respectively).

Model	EE	TE	0-100	100-500	500+	All
RC (random initializations)	44.5	64.4	40.5	57.8	63.4	53.4
RC (SG initializations)	49.5	68.6	44.1	62.5	67.0	57.3
RC (SG fixed)	48.9	**68.7**	44.1	62.7	67.3	57.6
RC + SG	51.6	67.4	**46.4**	62.5	66.8	58.2
RC + SGLR	**51.7**	68.5	45.3	**63.0**	**68.1**	**58.4**

best for the higher frequency ranges. This can be explained by the fact that the SGLR separates left and right context words, creating sparser and more precise contexts compared to SG. Sparse context descriptions can hurt representations of low frequency words as there may not be enough words that share contexts. But, for more frequent words, more precise context descriptions as in SGLR help to prevents incorrect generalizations (such as cases where word order matters). When evaluating on the full Dev set, both combined loss settings outperform the baselines consistently.

5.4 Comparison to the State of the Art

We also compared our proposed models to various state-of-the-art systems from the literature:

The THYME system, by Lin et al. (2016), consist of separate models for EE relations and TE relations. They employ two feature rich support vector machines (SVM), using POS and dependency parse features from the cTAKES clinical pipeline (Savova et al., 2010) together with augmented training through extended UMLS entities. They later replaced the TE component by a token-based CNN model which improved their model (Lin et al., 2017; Dligach et al., 2017). Also replacing the EE component by a CNN model decreased model performance, showing that the CNN was not able to replace the feature rich SVM. Leeuwenberg and Moens (2017) used a feature rich structured perceptron, also using cTAKES POS and dependency parse features, jointly learning different relation types on the document level. Tourille et al. (2017) used two bidirectional LSTM models, one for inter-sentence and one for intra-sentence relations. They used fixed word embeddings pre-trained on the MIMIC III corpus, and also incorporated character level information. To obtain their top results they added ground truth event attribute features enhanced with entity information also obtained from cTAKES.

As can be noticed, all state-of-the-art baselines used dedicated clinical NLP tools to enhance their features in order to obtain their top results, in contrast to our model, which uses only the Stanford POS Tagger (trained on news texts).

Table 2 shows that initializing the model with the pre-trained embeddings gives a significant [4] 1.1 point increase in F-measure compared to random initialization, due to an increase in precision. Fixing the embeddings gives slightly better performance than using them as initialization, an increase of 0.9 point in F-measure, mostly due to higher recall. When extending the loss with the SGLR loss, we gain[6] 1.6 in F-measure compared to fixing the word embeddings, and also surpass the state of the art by 0.4 even without specialized resources. If we train our model using the SG loss extension we obtain the best results, and gain[6] 1.9 points in F-measure compared to using pre-trained fixed word embeddings. This setting also exceeds the state of the art (Lin et al., 2017) by 0.7 points in F-measure, due to a gain of 1.2 points in recall, again without using any specialized clinical NLP tools for feature engineering, in contrast to all state-of-the-art baselines.

5.5 Manual Error Analysis

Finally, we manually analyzed 50 false positives and 50 false negatives picked randomly from the test set predictions for different settings.

[4] $P < 0.0001$ for a document-level pairwise t-test

Table 2: THYME test set results, reporting precision (P), recall (R) and F-measure (F), macro-averaged over three runs. The standard deviation for F is also given.

Model	P	R	F
With specialized resources:			
Best Clinical TempEval (2016)	58.8	55.9	57.3
Lin et al. (2016)	66.9	53.4	59.4
Leeuwenberg et al. (2017)	-	-	60.8
Tourille et al. (2017)	65.7	57.5	61.3
Lin et al. (2017)	66.2	58.5	62.1
No specialized resources:			
RC (random initialization)	67.9	52.1	58.9$_{\pm 0.2}$
RC (SG initialization)	**71.2**	52.0	60.0$_{\pm 1.2}$
RC (SG fixed)	68.9	54.6	60.9$_{\pm 0.8}$
RC + SG	66.2	**59.7**	**62.8**$_{\pm 0.2}$
RC + SGLR	68.7	57.5	62.5$_{\pm 0.3}$

From Table 3 we can see that all models have difficulties with distant relations that cross sentence or clause boundaries (CCR). This could be because class imbalance correlates with distance between the arguments of the temporal relations. Furthermore, arguments that are frequent in the supervised data (> 250) are a dominant error category. We suspect this is because frequent events often function both as container and as contained, whereas infrequent events are less ambiguous in their argument position. This hurts RC (SG fixed) most as its embeddings are not influenced by D^{rc}. Furthermore it can be noticed that RC+SG has less errors for infrequent arguments (< 10) in the supervised data. This could be because it leverages the few available instances from both the D^{rc} and D^{sg} data better than the single-loss models.

Table 3: Error analysis on 50 FP and 50 FN (random from test) for different settings. Clause boundaries are: newlines and sub-clause or sentence boundaries. Error categories are not mutually exclusive.

Error Type	RC + SG	RC (SG fixed)	RC (SG init.)
Cross-Clause Relations (CCR)	42	39	36
Infrequent Arguments (< 10)	11	15	26
Frequent Arguments (> 250)	37	50	40
Mistake in Ground-Truth	10	8	5
Other	21	15	28

6 Conclusions

In this work, we proposed a neural relation classification model for the extraction of narrative containment relations from clinical texts.[5]

The model trains word representations jointly on the supervised relation classification task and an unsupervised auxiliary skip-gram objective (with separate datasets) through weight sharing to more effectively exploit both the unlabeled and labeled data, as annotated clinical data is costly to create. We show that this word-level joint training results in significantly better generalizing classification models compared to using pre-trained word embeddings (either as initialization or fixed embeddings). Furthermore, we show that performance trends and good values for λ (balance between tasks) are robust over different training set sizes, and that even for (badly tuned) extreme values of λ the quality of the model's

[5] Code is available at: `http://liir.cs.kuleuven.be/software.html`

embeddings is naturally lower-bounded by their pre-trained variants. Additionally, our model sets a new state of the art for temporal relation extraction on the THYME dataset, without using extra dedicated clinical resources, in contrast to current state-of-the-art models.

As future work, it would be interesting to see how well the improvements caused by the word-level joint training generalize to other NLP tasks that typically use pre-trained word embeddings.

Acknowledgment

The authors would like to thank the reviewers for their constructive comments which helped us to improve the paper. Also, we would like to thank the Mayo Clinic for permission to use the THYME corpus. This work was funded by the KU Leuven C22/15/16 project ”MAchine Reading of patient recordS (MARS)”, and by the IWT-SBO 150056 project ”ACquiring CrUcial Medical information Using LAnguage TEchnology” (ACCUMULATE).

References

Jonathan Baxter et al. 2000. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12(149-198):3.

Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. SemEval-2015 task 6: Clinical TempEval. In *Proc. of SemEval*, pages 806–814. ACL.

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 task 12: Clinical TempEval. *Proc. of SemEval*, pages 1052–1062.

Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovsky. 2017. SemEval-2017 task 12: Clinical TempEval. In *Proc. of SemEval*, pages 565–572, Vancouver, Canada, August. ACL.

Rich Caruana. 1998. Multitask learning. In *Learning to Learn*, pages 95–133. Springer.

Hao Cheng, Hao Fang, and Mari Ostendorf. 2015. Open-domain name error detection using a multi-task RNN. In *Proc. of EMNLP*, pages 737–746.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Leon RA Derczynski. 2017. Automatically ordering events and times in text. In *Studies in Computational Intelligence*, volume 677. Springer.

Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. *Proc. of EACL*, page 746.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proc. of ACL*, pages 1723–1732. ACL.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. *Proc. of EMNLP*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Min Jiang, Yang Huang, Jung-wei Fan, Buzhou Tang, Josh Denny, and Hua Xu. 2015. Parsing clinical text: how good are the state-of-the-art parsers? *BMC Medical Informatics and Decision Making*, 15(1):S2.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, 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.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Hee-Jin Lee, Hua Xu, Jingqi Wang, Yaoyun Zhang, Sungrim Moon, Jun Xu, and Yonghui Wu. 2016. Uthealth at SemEval-2016 task 12: an end-to-end system for temporal information extraction from clinical notes. In *Proc. of SemEval*, pages 1292–1297. ACL.

Artuur Leeuwenberg and Marie-Francine Moens. 2017. Structured learning for temporal relation extraction from clinical records. In *Proc. of EACL*, pages 1150–1158. ACL.

Chen Lin, Dmitriy Dligach, Timothy A Miller, Steven Bethard, and Guergana K Savova. 2015. Multilayered temporal modeling for the clinical domain. *Journal of the American Medical Informatics Association*, 23(2):387–395.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2016. Improving temporal relation extraction with training instance augmentation. In *Proc. of ACL*, page 108. ACL.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2017. Representations of time expressions for temporal relation extraction with convolutional neural networks. In *Proc. of BioNLP*, page 322. ACL.

Wang Ling, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proc. of HLT-NAACL*, pages 1299–1304. ACL.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nelson Morgan and Hervé Bourlard. 1990. Generalization and parameter estimation in feedforward nets: Some experiments. In *Advances in Neural Information Processing Systems*.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proc. of NAACL-HLT*, pages 39–48.

Agnieszka Onisko, Allan Tucker, and Marek J. Druzdzel. 2015. Prediction and prognosis of health and disease. In *Foundations of Biomedical Knowledge Representation: Methods and Applications*, pages 181–188. Springer.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 548–554.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proc. of LREC*, pages 2089–2096. European Language Resources Association (ELRA), May.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proc. of ACL*, volume 2, pages 231–235. ACL.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

William F Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, et al. 2014. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

Julien Tourille, Olivier Ferret, Aurelie Neveol, and Xavier Tannier. 2017. Neural architecture for temporal relation extraction: A Bi-LSTM approach for detecting narrative containers. In *Proc. of ACL*, pages 224–230, Vancouver, Canada, July. ACL, ACL.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of NAACL-HLT*, pages 173–180. ACL.

Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. 2012. Tempeval-3: Evaluating events, time expressions, and temporal relations. *arXiv preprint arXiv:1206.5333*.

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proc. of EMNLP*, pages 236–246. ACL, November.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *Proc. of COLING*, pages 2335–2344. ACL.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proc. of PACLIC*.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proc. of ACL*.