A Hybrid Knowledge and Transformer-Based Model for Event Detection with Automatic Self-Attention Threshold, Layer and Head Selection

Thierry Desot, Orphée De Clercq, and Veronique Hoste
LT[3], Language and Translation Technology Team
Ghent University, Groot-Brittanniëlaan 45, 9000 Gent, Belgium
{thierry.desot,orphee.declercq,veronique.hoste}@ugent.be

Abstract

Event and argument role detection are frequently conceived as separate tasks. In this work we conceive both processes as one task in a *hybrid* event detection approach. Its main component is based on *automatic keyword extraction* (AKE) using the self-attention mechanism of a *BERT* transformer model. As a bottleneck for AKE is defining the threshold of the attention values, we propose a novel method for *automatic self-attention threshold selection*. It is fueled by core event information, or simply the verb and its arguments as the backbone of an event. These are outputted by a knowledge-based syntactic parser. In a second step the event core is enriched with other semantically salient words provided by the transformer model. Furthermore, we propose an *automatic self-attention layer and head selection mechanism*, by analyzing which self-attention cells in the BERT transformer contribute most to the hybrid event detection and which linguistic tasks they represent. This approach was integrated in a pipeline event extraction approach and outperforms three state of the art multi-task event extraction methods.

1 Introduction

Event extraction, argument and semantic role detection are frequently conceived as separate tasks (Ji and Grishman, 2008; Gupta and Ji, 2009; Hong et al., 2011; Chen et al., 2015; Nguyen and Grishman, 2015; Liu et al., 2016a,b) where a *multi-word* event is first split into a verb as *single-word* event to process, after which its argument roles (subject, direct and indirect object(s)) and semantic roles (such as time and location) are extracted. These are typically trained in a multi-task setup for *event extraction*, which combines event span detection and classification. In this work, we tackle *multi-word* event extraction and conceive event span detection and argument extraction as *one task* in a hybrid knowledge and transformer-based event detection method. The verb, subject and object(s) (SVO) are first outputted by a knowledge-based syntactic parser and combined with *automatic keyword extraction* (AKE). In this latter step, the most relevant keywords in a sentence or most salient semantic information is selected, exploiting the attention mechanism of a transformer, i.e., *BERT* (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018). A bottleneck for AKE is defining the threshold of the attention values to take into account (Tang et al., 2019). Hence, we propose and outline a method for *automatic attention threshold selection* by exploiting the interaction between self-attention based AKE and rule-based event detection. As the main function of the rule-based component is to provide the necessary information for the automatic attention threshold mechanism, it targets only minimal event information, i.e., the core or backbone of the event or the verb and its SVO arguments. This allows the transformer's main component to complement it with other semantic roles and semantically salient information. However, the latter type of information is often essential to constitute the core meaning of the event. For example, omitting the adverb *"conditionally"* in the event *"He was conditionally released from detention."* changes its semantics and causes a misunderstanding of it. This kind of semantically salient information can only be provided by the transformer model, and not by the knowledge-based component in our hybrid model.

Our hybrid event detection mechanism is embedded in a *pipeline* event extraction approach that goes beyond short event spans: in a first step, event classification is applied to raw input sentences, whereas in a second step, the event span is detected. For a fair evaluation, we compare this approach with three event detection approaches as part of a multi-task event extraction method that jointly predicts event spans and classes. The main contributions of this paper are the following:

Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE), pages 21 - 31
December 7-8, 2022 ©2022 Association for Computational Linguistics

- To the best of our knowledge, this is the first work on hybrid event detection that conceives event span detection and argument extraction as one task. On top of that, AKE is integrated and combined with a novel automatic attention threshold selection mechanism.

- We also propose an automatic self-attention layer and head selection mechanism by investigating which layers and heads of the BERT transformer model contribute most to event detection, and which linguistic tasks they perform. Identifying such tasks in the transformer model can contribute to the creation of more domain-specific and tailor-made BERT models. Our methodology is language-independent. All experiments have been conducted on a Dutch corpus only, mainly because we did not find data in other languages with similar event *prominence* annotations (Section 3.1).

Our approach is positioned with respect to the state of the art in Section 2 and is presented in Sections 3 and 4. An overview of the data is given in Section 5. Section 6 presents the results of experiments, followed by a thorough analysis and discussion. The paper is concluded in Section 8.

2 Related Work

Knowledge-based event detection methods were initially based on ontologies (Frasincar et al., 2009; Schouten et al., 2010; Arendarenko and Kakkonen, 2012) or rule-sets (Valenzuela-Escárcega et al., 2015) which represent expert knowledge. These also include extracting candidate event words with part-of-speech tags (Mihalcea and Tarau, 2004), which can also satisfy predefined syntactic patterns (Nguyen and Phan, 2009). Statistical methods spot event spans using n-grams (Witten et al., 2005; Grineva et al., 2009), term frequency inverse document frequency (TF-IDF), word frequency and word co-occurrence (Kaur and Gupta, 2010).

Early supervised machine learning approaches recast event detection as a binary classification problem (Hasan and Ng, 2014) to decide whether an input word is part of an event or not. To that end, maximum entropy (Yih et al., 2006), support vector machines (SVM) (Lopez and Romary, 2010) and conditional random fields (CRF) (Zhang, 2008) were applied. As the event detection field initially concentrated on *fixed* event types using *single-word* or event spans with a short length (Mitamura et al., 2015), these supervised machine learning approaches have successfully used the ACE 2005 corpus (Walker et al., 2006) comprising *single-word* event span length annotations. With *feature engineering* approaches emerging, the scope became larger than a one-word event span (Patwardhan and Riloff, 2009). In Lefever and Hoste (2016) *multi-word* events in Dutch news text are detected using an SVM binary classifier combining lexical, syntactic and semantic features. These feature-based machine learning techniques, however, have been superseded by deep learning techniques which are able to learn hidden feature representations automatically from data. In Wang et al. (2017), a multi-word event detection approach using convolutional neural networks (CNN) outperforms an SVM approach. Spearheaded by their success in dealing with long-term dependencies in longer sequences, the LSTM (long short-term memory) and attention mechanism allow the decoder to learn which parts of the sequence should be attended to in an encoder-decoder architecture (Bahdanau et al., 2014; Luong et al., 2015), hence taking more context information into account. Zhao et al. (2018) presents a supervised *attention-based* RNN event detection approach that outperforms an RNN and CNN, both without attention mechanism.

Deep learning approaches that were in recent years combined with Word2Vec (Mikolov et al., 2013), GLoVe (Pennington et al., 2014) and fastText (Bojanowski et al., 2017) word embeddings have led to the rise of the *transformer architecture* (Vaswani et al., 2017). Its *contextual language models* have been successfully integrated in a range of NLP tasks using pre-trained contextual *BERT* (Bidirectional Encoder Representations from Transformers) word embeddings (Devlin et al., 2018). On top of that, the BERT model fully exploits the attention mechanism for multi-word event detection, which is illustrated in Mehta et al. (2020), where a multi-attention event detection tool, using BERT, fine-tuned on the Civil Unrest Gold Standard Report data (Ramakrishnan et al., 2014), outperforms a CNN. The *hybrid* target event detection method that is proposed here also fully benefits from the BERT multi-head self-attention, but is combined with subject, verb and object (*SVO*) information, as outputted by a knowledge-based syntactic parser.

Figure 1: Example of EventDNA corpus event spans and Main, Background, None event prominence labels

3 Pipeline Event Extraction Approach

An event can be defined as the smallest extent of text that expresses its occurrence (Song et al., 2015) and is identified by a word or phrase called event *trigger, nugget, event span or mention*. Event mentions can be *single-word* event triggers that are usually (main) verbs, nouns, adjectives and adverbs. *Multi-word* event triggers can be consecutive tokens, complete sentences, or *discontinuous* when on top of the verb, its participants, or argument roles are also involved (Doddington et al., 2004). Our *hybrid* event detection approach targets *multi-word continuous event* spans. It goes *beyond* the scope of approaches tackling *single-word* events that are frequently using the ACE 2005 corpus (Section 2). Hence our models are trained on the (Dutch) EventDNA corpus, annotated with multi-word event spans and class labels (Section 5).

3.1 Event Prominence Classification

Our hybrid model is part of a pipeline event extraction model which comprises an event classifier and detection module. Event prominence classification was chosen, other than the typically used event type classification (Desot et al., 2021) that frequently fails to handle the variety of events expressed in real-world situations. To overcome this, we classify new information into *prominence* classes. Hence, the input sentence can be classified as `Main` event when it exhibits new information and, for example in a news context actually caused the reporter to write the article; or as `Background` event when it gives context or background to the `Main` event. Raw sentences without events are classified as `None` events. Figure 1 presents an example of an event span labeled as `Background` event, preceded by a `Main` and `None` event.

For event classification a transformer-based BERT model for the Dutch language, BERTje (de Vries et al., 2019) has been pre-trained on a dataset of 2.4 billion tokens from Wikipedia, Twente News Corpus (Ordelman et al., 2007) and SoNaR-500 (Oostdijk et al., 2013) with masked language modeling and next sentence prediction. BERTje has an architecture of 12 transformer blocks (bidirectional layers) and 12 self-attention heads and a hidden size of 768. This Dutch language model has been fine-tuned for sequence (event) classification on the raw sentences of the EventDNA data set (Section 5). Only output sentences with predicted `Main` prominence class or `Background` class are accepted as input for hybrid knowledge- and transformer-based event detection, whereas sentences predicted as `None` events are not further processed.

3.2 Knowledge and Transformer-Based Event Detection with Automatic Attention Threshold Selection

The main function of the rule-based part of our *hybrid* event detection approach is to provide the necessary information to the automatic attention threshold selection mechanism. Hence, the backbone of the event, i.e., the subject (`SUBJ`), (head) verb (`VERB`) and object (`OBJ`) information is outputted by a knowledge-based syntactic parser for the Dutch language, namely Alpino. This parser combines a rule-based head-driven phrase structure grammar (HPSG) with a lexicon of 100,000 entries and a part-of-speech (POS) tagger. On top of that, dependency parse trees are generated, which are disambiguated with a maximum entropy component (Van der Beek et al., 2002; Van Noord et al., 2006; Smessaert and Augustinus, 2010). For this parser, an F1 score of 91.14% has been reported on 1,400 manually annotated sentences from the Twente News corpus (Ordelman et al., 2007). Starting from these predicted tags a set of rules is then used to align them with the corresponding words.

In a next step, the syntactic output is the cornerstone of our automatic attention threshold selection mechanism. To this purpose, *automatic keyword extraction* (AKE) exploiting the attention mechanism of BERTje (Section 3.1) is used. Keywords are defined as the most relevant words in an event span (Sarracén and Rosso, 2021), and are extracted through attention weights obtained over the 12 x 12 transformer self-attention layers and heads from the BERTje model. In the study of Tang et al. (2019) only 10% of the words with the highest attention weights were kept as keywords. Initially, and in a similar vein, given a sequence of attention weights in $Att = (Att_1, ..., Att_n)$, in ascending weight value order, we identified the words above a certain threshold. We iteratively explored a range of threshold values between 0.1 and 0.9 per step of 0.1 to find an optimal threshold (0.25). This was only a preparatory step in order to estimate the feasibility of our approach, as such a *fixed* threshold percentage is arbitrary and not optimal over sentences with different lengths and data sets. Hence, we defined an *automatic* and *variable* threshold (Att_{thresh}) as the minimum value for the attention values to be selected. To this purpose the percentage (p) of subject, verb, object words ($\#SVO\_words$), as output from the previous step, in relation to the total number of words per sentence ($\#Sentence\_words$) was calculated as $p = \frac{\#SVO\_words*100}{\#Sentence\_words}$. The threshold is the *lowest* attention weight in the range of the p percent of the top-ranked attention values per sentence, which we calculate using the $percentile$, $Att_{thresh} = percentile(Att, (100 - p))$. Finally, the resulting top-ranked attention values Att exceeding the threshold Att_{thresh} are selected (Att_{sel}), where $Att_{sel} = (Att_{thresh}, ..., Att_n)$. The subtokens of the words corresponding to these values are kept as keywords, discarding the special separator `[SEP]` and classification tokens `[CLS]`. The subtokens are then again concatenated into words. With the BERTje model, not all subtokens of a word have equal attention weights. In that case, we extracted the whole word as keyword if one of its subtokens passes the threshold. The resulting attention-based keywords are merged with the (*SVO*) combinations of the event detection module. Finally, the original word order is restored by aligning the merged words with the original input sentence. Figure 2 depicts the complete event detection process for the Dutch input sentence "(The company) XYZ moet extra personeelsleden vinden wegens uitval van werknemers."[1]

We want to emphasize that other argument roles, such as time and place on top of the SVO words, were not considered and are not outputted by the knowledge-based parser. In our initial experiments, these resulted in a too high percentage of selected words and too low threshold values, which led to an overgeneration of predicted event words. However, part of these semantic roles do occur in the semantically salient words predicted by the transformer model (Section 7).

3.3 Automatic Self-Attention Layer and Head Selection

Certain self-attention layers and heads of the transformer model exhibit linguistic notions, such as syntax and coreference (Vig, 2019; Vig and Belinkov, 2019; Clark et al., 2019). According to several studies (Goldberg, 2019; Hewitt and Manning, 2019; Jawahar et al., 2019; Vig and Belinkov, 2019) on the BERT transformer, attention follows syntactic *dependency* and subject-verb-object agreement most strongly in the *middle layers* of the BERT model. In order to automatically select the self-attention layer and head that contribute most to event detection performances, we exploit the interaction between the transformer and knowledge-based syntactic parser again and verify the number of SVO words predicted by the transformer model. We first apply our automatic threshold selection technique per self-attention transformer cell by calculating the attention values per isolated head per layer (Vig and Belinkov, 2019), for each of the 12 x 12 transformer cells (144 times) on the test data (Section 5). In a next step, per transformer matrix cell we calculate the percentage of overlap between selected event tokens with an attention value above the automatically selected threshold and between the knowledge-based predicted SVO words. We finally consider the self-attention layer and cell that output most SVO words, as exhibiting the linguistic notion of syntactic dependency. We verify if it improves event detection performance and analyse the behaviour of this layer in Section 7.

4 Baseline Multi-Task Event Extraction Approaches

We compare the target *pipeline* event extraction model (Section 3) with three baseline multi-task event extraction models. To the best of our knowledge, we are not aware of other baseline approaches applied to languages with a similar event *prominence* annotation scheme (Section 3.1). In the multi-task approach, event detection and classification tasks are performed simultaneously to benefit from their interplay (Li et al., 2013; Liu et al., 2017). The first model is an attention-based RNN model with LSTM from Liu and Lane (2016), with an encoder-decoder architecture. Its atten-

[1]English translation: *"(The company) XYZ has to find extra staff due to employee absence."*

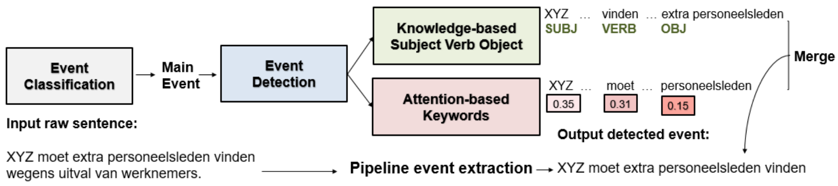

Figure 2: Overview of pipeline event extraction

Event span IOB labels:									**Event class:**
B-EV	I-EV	I-EV	I-EV	I-EV	O	O	O	O	Main
Raw input sentence									
XYZ	moet	extra	personeelsleden	vinden	wegens	uitval	van	werknemers.	

Table 1: Input raw sentence with event detection IOB labels and class

tion context vector provides information from parts of the input sequence that the classifier pays attention to. The second model is fine-tuned for combined event detection and classification on the same pre-trained BERTje model as our target approach (Section 3.1). For combining both tasks, given the input token sequence $x = (x_1, ..., x_T)$, the output hidden states of the BERTje model are $H = (h_1, ..., h_T)$. For event detection the final hidden states of $(h_2, ..., h_T)$ are fed into a softmax layer to classify over the detected event subtokens s. Based on the hidden state of the (first) special classification `[CLS]` token, denoted as h_1, the event y with weighted representations of query, key and value vectors W is predicted as,

$$y_n^s = softmax(W^s h_n + b^s), n \in 1 \dots N \quad (1)$$

and the detected event sequence as $y^s = (y_1^s, ..., y_T^s)$ which are then jointly modeled as,

$$p(y^i, y^s|x)) = p(y^i|x) \prod_{n=1}^{N} p(y_n^s|x) \quad (2)$$

which maximizes the probability $p(y^i, y^s|x))$.

We finally added a CRF on top of the multi-task BERTje-based approach, resulting in our third baseline model where the joint BERT+CRF replaces the softmax classifier with CRF (Chen et al., 2019). The target event sequence is labeled in *IOB* format. Tokens at the *begin* of an event mention are labelled as *B-EV*, tokens *inside* the mention as *I-EV*, and tokens *outside* the mention as *O*. Table 1 includes the same example sentence as in Figure 2.

5 Data

Both event extraction approaches (Sections 3 and 4) were trained and tested using the event span and label annotations in the titles and lead paragraphs of the EventDNA corpus. This corpus comprises news articles and follows the *ERE* (Entities, Relations, Events) annotation standards (Song et al., 2015; Aguilar et al., 2014). For more detailed information about the corpus we refer the reader to Desot et al. (2021) which outlines event classification experiments, for validating the quality of the corpus and to Colruyt et al. (2019, accepted for publication) for the corpus design and annotations. A high number (32%) of event types in the EventDNA corpus do not correspond to the event types specified in the ERE-based EventDNA annotation protocol. Hence, event prominence classification was chosen, other than the typically used event type classification (Desot et al., 2021), as explained in Section 3.1 and Figure 1.

Events	**#**	**Item**	**#**
Main	4175	Vocab.	13050
Backgr.	3100	Tokens	88530
None	1792	Sentences	6813
Total	9069	Documents	1740

Table 2: Overview of EventDNA corpus statistics

The EventDNA data set comprises raw sentences with more than one event span. As a first step, only unique sentences with one event span were kept for our experiments. Table 2 exhibits information about the data set used for our experiments (Section 6), with an overview of the event prominence class distributions (first column). In order to train our models, the (6813) sentences of the data set were split into 80% train, 10% development (*Dev.*) and 10% held-out test partitions.

6 Experiments and Results

6.1 Baseline Multi-Task Event Extraction

The *raw sentences* in the training data set were used to train the baseline multi-task models and was automatically converted into *IOB* format (Section 4). The attention-based RNN model was trained for 10 epochs with a batch size of 10, using Adam optimizer, and with the number of LSTM cell units set as 128. Word embeddings of size 128 were randomly initialized. For fine-tuning the BERT-based models, optimal performances were obtained using the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 1e-5 and a batch size of 10 instances during 10 epochs. The maximum sequence length is set to 82 tokens, which is the maximum sequence token length of the training data sentences. The special `[CLS]` (classification) token and `[SEP]` (separator) tokens were inserted.

Table 3 shows that the attention-based RNN model (*Att-RNN.*) is outperformed by the BERT-based models. The combined BERTje and CRF multi-task model (*BERTje+CRF*) outperforms the BERTje model without CRF (*BERTje*) for both event detection and classification. We compared event detection with (Table 3, *+class.*) and without (*-class.*) interaction with classification. Multi-task event detection benefits from the interaction between event classification and detection and outperforms event detection without the impact of event classification.

6.2 Target Pipeline Event Extraction

The target pipeline event extraction approach is composed of a BERTje-based *classifier* and a hybrid knowledge- and transformer attention-based *event detection* approach. Raw sentences that are classified as `Main` and `Background` events are fed to the hybrid event detection tool in order to identify the event span in the raw sentence. Similar parameters as used for the BERTje-based multi-task baseline models (Section 6.1) have been applied, except for a lower number (3) of epochs in order to obtain optimal performances. *Event prominence classification* performance on the test set is exhibited in Table 4, *Event class*, which outperforms classification of the baseline multi-task models (Table 3). As a next step, the sentences classified as `Main` or `Background` event, are fed to the hybrid *event detection* module that combines rule-based extraction of *SVO* words with self-attention based extraction of keywords. Performances in Table 4 are compared for:

- a fixed self-attention threshold (Section 3.2), *Fix. thresh.* of 0.25
- automatic self-attention threshold selection (Section 3.2), *Aut. thresh.*
- combined self-attention threshold, layer and head selection (Section 3.3), *Aut. thresh. + layer*.

These performances were calculated for raw sentence words, predicted as *inside*, *outside*, or in *initial* position of the gold standard annotated event spans of our data set. The model with a fixed threshold (*Fix. thresh.*) outperforms the second attention model with an automatically selected threshold (*Aut. thresh.*), although performances for the latter model are methodologically more fair. Performances on the gold standard event classes (*Fix. thresh. Gold.*) are slightly better compared to detection of events for the predicted event classes (*Fix. thresh. Pred.*). Best results however are shown for automatic threshold combined with self-attention layer and head selection (*Aut. thresh. + layer*) (layer 7, head 1). Event detection was also performed using attention-based keywords without knowledge-based predicted words (*Att.*) and vice versa (*SVO*). These results demonstrate that event detection performance increases, if knowledge- and attention-based event detection are combined.

7 Results Analysis and Discussion

In spite of the interaction between event classification and event detection, the multi-task baseline models could not outperform the classifier of the target pipeline model. On top of that, the pre-trained BERTje models of the BERT-based multi-task baseline models outperform the attention-based RNN multi-task model without BERT. This shows that a pre-trained BERT transformer model improves performances, when fine-tuning on a small data set. For event detection with automatic self-attention threshold selection, the target pipeline event extraction model did not outperform the BERT-based baseline models. However, combined with automatic self-attention layer and head selection, layer 7 and head 1 show the best event detection performances.

Hence, we analysed the latter result by correlating the order of transformer block layers and heads

Event extraction	Event classification			Event detection			Class
	Prec.	Rec.	F1	Prec.	Rec.	F1	+/-
Baseline multi-task models:							
Att.-RNN	0.52	0.54	0.52	0.60	0.61	0.60	+class
	-	-	-	0.58	0.59	0.58	-class
BERTje	0.60	0.61	0.60	0.66	0.65	0.65	+class
	-	-	-	0.65	0.64	0.64	-class
BERTje+CRF	0.61	0.62	0.61	0.66	0.67	0.66	+clas s
	-	-	-	0.65	0.65	0.65	-class

Table 3: Overview of baseline multi-task event extraction performances

Event extraction		Prec.	Rec.	F1
Event class.		0.69	0.68	**0.68**
Event det.				
Fix. thresh.	Gold.	0.83	0.57	0.65
	Pred.	0.79	0.58	0.64
Aut. thresh.		0.75	0.57	0.63
	SVO	0.70	0.51	0.57
	Att.	0.71	0.54	0.60
Aut. thresh. + layer		0.88	0.62	**0.71**

Table 4: Pipeline model event extraction performances

Correlation	Pearson	Spearman
Layer order	-0.30*	-0.36*
Head order	-0.13**	-0.12**

$^{*}p < 0.05$; $^{**}p > 0.05$

Table 5: Layer/head order - event detection correlation

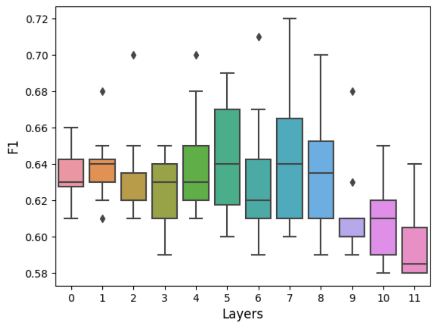

Figure 3: Transformer self-attention layer depth and hybrid target model event detection F1 scores

Figure 4: Hybrid event detection F1 score - overlap self-attention and knowledge-based model output SVO tokens per self-attention layer

with event detection F1 scores. In a next step, attention attributions of the transformer model are visualised. Finally we check the impact on attention attribution stability by changing the word order of the input sentences.

7.1 Correlation between Transformer Layers, Heads and Event Detection Performances

Pearson's correlation coefficient was calculated, measuring the association strength between two variables and *Spearman's rank correlation* that measures correlations between two *ranked variables*. We use the p-value to determine if the resulting correlation coefficient is significant and whether or not to reject a null hypothesis. We reject the null hypothesis if the p-value is less than 0.05 ($p < 0.05$). Table 5 demonstrates *weak*, but *significant* ($p < 0.05$) negative Pearson and Spearman's rank correlations, -0.3 and -0.36 respectively, between event detection F1 scores and layer depth, unlike correlations between F1 scores and attention *heads*, which are not significant ($p > 0.05$). Figure 3 presents F1 scores (*F1*) averaged over the (12) heads per layer and shows a downward trend for F1 scores: maximum F1 score is obtained for middle *layer* 7 (0.71), whereas the minimum F1 scores are shown for the deepest layers 10 and 11. A similar trend is shown in Figure 4. It presents the percentages in overlap between the knowledge-based predicted SVO words and event tokens with an attention value above the automatically selected threshold (averaged over the 12 attention heads per layer), which we calculated for automatic self-attention layer and head selection (Section 3.3).

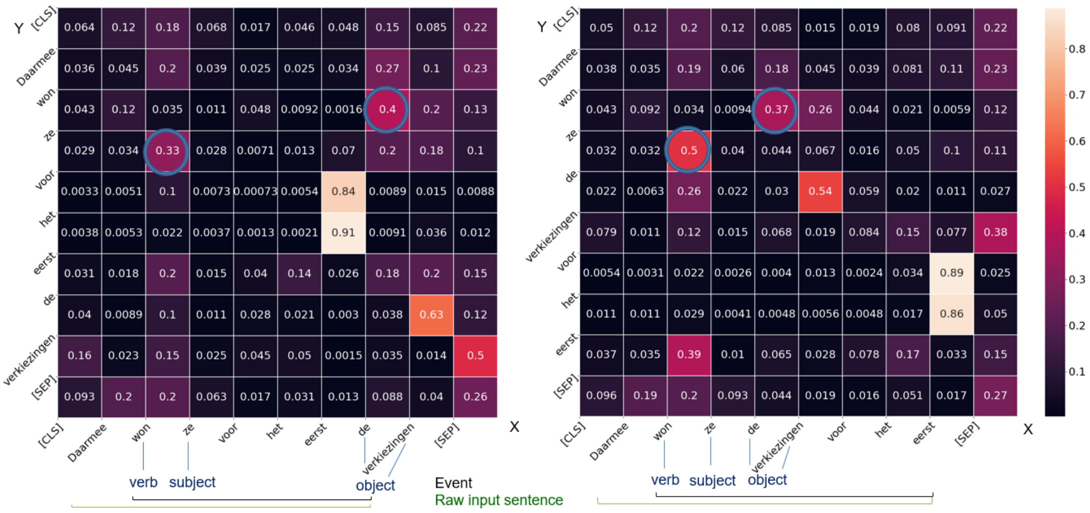

Figure 5: Self-attention values for SVO dependencies, layer 7 and head 1, without and with changed word order

The highest overlap is shown for layer 7, resulting in the best event detection *F1* scores (normalized to percentage). This indicates that layer 7 can be identified most with the notion of SVO dependencies. Furthermore, correlations in Table 5 show that layers are associated more with linguistic reasoning tasks than heads. This supports the hypothesis in the study of Hoover et al. (2019) that dependencies are probably encoded by a combination of heads rather than by a single head.

7.2 Attention Attribution and Stability

As attention follows SVO agreement most strongly in layer 7, head 1 of the BERTje model we visualise these attentions for the test set. For 100 randomly selected test sentences, with SVO attention values above the automatically selected threshold, we changed the word order (without changing the meaning). For the resulting sentences we found that for 61%, the same dependencies and words with most attention are preserved. This indicates a consistent behaviour of the BERTje model w.r.t. attention attributions. For the Dutch sentence *"And so she won the elections for the first time."*[2], the circles in the left attention heatmap matrix (Figure 5) mark intersections in cells with a high attention value that show a dependency between the `verb` (*"won"*) and `object` (*"de verkiezingen"*), and between the `subject` (*"ze"*) and the `verb` (*"won"*) with their corresponding words on the X and Y axis. Among the keywords with a weight $>$ the threshold, the keyword with most attention (0.91) is *"eerst"*[3] in the collocation *"voor het eerst"* [4], a semantically very salient word in this context. *"voor het eerst"*, was moved to the end of the event (Figure 5, right heatmap), and has still the highest attention value (0.89), with the same SVO dependencies.

8 Conclusion and Future Work

This study outlines a pipeline hybrid knowledge- and transformer self-attention based event detection approach. It outperforms three state of the art multi-task baseline event extraction models. For keyword-based event detection, we solved the bottleneck of defining the threshold of the attention values to take into account. Automatic self-attention threshold, layer and head selection was applied, exploiting the interaction between a rule-based SVO (subject-verb-object) extraction and self-attention based automatic keyword extraction (AKE). Analysis of the BERTje transformer model shows that syntactic dependencies are most active in the middle layers and contribute most to event detection. We also found evidence for consistency of attention attributions of the transformer model. As a next step, the behaviour and stability of the surrounding layers, should be further investigated. Other data sets in Dutch or other languages can be used, comprising more than one event span per sentence.

Acknowledgements

This work was supported by Ghent University under grant BOFGOA2018000601.

[2]Original Dutch sentence: *"Daarmee won ze voor het eerst de verkiezingen."*

[3] *"first"*

[4] *"for the first time"*

References

Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song, and Joe Ellis. 2014. A comparison of the events and relations across ace, ere, tac-kbp, and framenet annotation standards. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 45–53.

Ernest Arendarenko and Tuomo Kakkonen. 2012. Ontology-based information and event extraction for business intelligence. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 89–102. Springer.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. 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)*, pages 167–176.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Camiel Colruyt, Orhpée De Clercq, Thierry Desot, and Veronique Hoste. accepted for publication. Eventdna: a dataset for dutch news event extraction as a basis for news diversification. *Language Resources and Evaluation*.

Camiel Colruyt, Orphée De Clercq, and Véronique Hoste. 2019. Eventdna: guidelines for entities and events in dutch news texts (v1. 0). *LT3 Technical Report-LT3 19-01*.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*.

Thierry Desot, Orphee De Clercq, and Veronique Hoste. 2021. Event prominence extraction combining a knowledge-based syntactic parser and a bert classifier for dutch. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 346–357.

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*.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Flavius Frasincar, Jethro Borsje, and Leonard Levering. 2009. A semantic web-based approach for building personalized news services. *International Journal of E-Business Research (IJEBR)*, 5(3):35–53.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multi-theme documents. In *Proceedings of the 18th international conference on World wide web*, pages 661–670.

Prashant Gupta and Heng Ji. 2009. Predicting unknown time arguments based on cross-event propagation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 369–372.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1262–1273.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 1127–1136.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2019. exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: Hlt*, pages 254–262.

Jasmeen Kaur and Vishal Gupta. 2010. Effective approaches for extraction of keywords. *International Journal of Computer Science Issues (IJCSI)*, 7(6):144.

Els Lefever and Véronique Hoste. 2016. A classification-based approach to economic event detection in dutch news text. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 330–335.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016a. Leveraging framenet to improve automatic event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2134–2143.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798.

Shulin Liu, Kang Liu, Shizhu He, and Jun Zhao. 2016b. A probabilistic soft logic based approach to exploiting latent and global information in event classification. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Patrice Lopez and Laurent Romary. 2010. Humb: Automatic key term extraction from scientific articles in grobid. In *SemEval 2010 Workshop*, pages 4–p.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Sneha Mehta, Mohammad Raihanul Islam, Huzefa Rangwala, and Naren Ramakrishnan. 2020. Interpretable event detection and extraction using multi-aspect attention.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

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

Teruko Mitamura, Zhengzhong Liu, and Eduard H Hovy. 2015. Overview of tac kbp 2015 event nugget track. In *TAC*.

Chau Q Nguyen and Tuoi Thi Phan. 2009. An ontology-based approach for key phrase extraction. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 181–184.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371.

Nelleke Oostdijk, Martin Reynaert, Véronique Hoste, and Ineke Schuurman. 2013. The construction of a 500-million-word reference corpus of contemporary written dutch. In *Essential speech and language technology for Dutch*, pages 219–247. Springer, Berlin, Heidelberg.

Roeland Ordelman, Franciska de Jong, Arjan Van Hessen, and Hendri Hondorp. 2007. Twnc: a multifaceted dutch news corpus. *ELRA Newsletter*, 12(3/4):4–7.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 151–160.

Jeffrey Pennington, Richard Socher, and Christopher D 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.

Naren Ramakrishnan, Patrick Butler, Sathappan Muthiah, Nathan Self, Rupinder Khandpur, Parang Saraf, Wei Wang, Jose Cadena, Anil Vullikanti, Gizem Korkmaz, et al. 2014. 'beating the news' with embers: forecasting civil unrest using open source indicators. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1799–1808.

Gretel Liz De la Peña Sarracén and Paolo Rosso. 2021. Offensive keyword extraction based on the attention mechanism of bert and the eigenvector centrality using a graph representation. *Personal and Ubiquitous Computing*, pages 1–13.

Kim Schouten, Philip Ruijgrok, Jethro Borsje, Flavius Frasincar, Leonard Levering, and Frederik Hogenboom. 2010. A semantic web-based approach for personalizing news. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 854–861.

Hans Smessaert and Liesbeth Augustinus. 2010. Nederbooms. *linguistics*, 2012.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ere: annotation of entities, relations, and events. In *Proceedings of the the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98.

Matthew Tang, Priyanka Gandhi, Md Ahsanul Kabir, Christopher Zou, Jordyn Blakey, and Xiao Luo. 2019. Progress notes classification and keyword extraction using attention-based deep learning models with bert. *arXiv preprint arXiv:1910.05786*.

Marco A Valenzuela-Escárcega, Gus Hahn-Powell, Mihai Surdeanu, and Thomas Hicks. 2015. A domain-independent rule-based framework for event extraction. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 127–132.

Leonoor Van der Beek, Gosse Bouma, Rob Malouf, and Gertjan Van Noord. 2002. The alpino dependency treebank. In *Computational linguistics in the netherlands 2001*, pages 8–22. Brill Rodopi.

Gertjan Van Noord et al. 2006. At last parsing is now operational. In *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, pages 20–42.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.

Anran Wang, Jian Wang, Hongfei Lin, Jianhai Zhang, Zhihao Yang, and Kan Xu. 2017. A multiple distributed representation method based on neural network for biomedical event extraction. *BMC medical informatics and decision making*, 17(3):59–66.

Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. 2005. Kea: Practical automated keyphrase extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, pages 129–152. IGI global.

Wen-tau Yih, Joshua Goodman, and Vitor R Carvalho. 2006. Finding advertising keywords on web pages. In *Proceedings of the 15th international conference on World Wide Web*, pages 213–222.

Chengzhi Zhang. 2008. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, 4(3):1169–1180.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 414–419.