Article

A Benchmark for Dutch End-to-End Cross-Document Event Coreference Resolution

Loic De Langhe *, Thierry Desot, Orphée De Clercq and Veronique Hoste

LT3, Language and Translation Technology Team, Ghent University, Groot-Brittanniëlaan 45, 9000 Ghent, Belgium
* Correspondence: loic.delanghe@ugent.be

Abstract: In this paper, we present a benchmark result for end-to-end cross-document event coreference resolution in Dutch. First, the state of the art of this task in other languages is introduced, as well as currently existing resources and commonly used evaluation metrics. We then build on recently published work to fully explore end-to-end event coreference resolution for the first time in the Dutch language domain. For this purpose, two well-performing transformer-based algorithms for the respective detection and coreference resolution of Dutch textual events are combined in a pipeline architecture and compared to baseline scores relying on feature-based methods. The results are promising and comparable to similar studies in higher-resourced languages; however, they also reveal that in this specific NLP domain, much work remains to be done. In order to gain more insights, an in-depth analysis of the two pipeline components is carried out to highlight and overcome possible shortcoming of the current approach and provide suggestions for future work.

Keywords: event coreference resolution; end-to-end; cross-document; Dutch language domain

check for updates

Citation: De Langhe, L.; Desot, T.; De Clercq, O.; Hoste, V. A Benchmark for Dutch End-to-End Cross-Document Event Coreference Resolution. *Electronics* **2023**, *12*, 850. https://doi.org/10.3390/electronics12040850

Academic editor: Min Zhang

Received: 13 January 2023
Revised: 2 February 2023
Accepted: 6 February 2023
Published: 8 February 2023

1. Introduction

End-to-end systems are critical in the development of real-world natural language processing (NLP) applications. The ability to directly extract valuable information or analyze raw texts using a single algorithm or a structured group of algorithms eliminates the need for cumbersome intermediary processing steps and the use of various systems that are often not optimally compatible. This in turn leads to systems that are both practical and streamlined, improving both their applicability and marketability [1]. As of now, many existing end-to-end applications are restricted to the English language domain, while other less-resourced languages are often overlooked. Yet, the growing global interest in NLP applications for text understanding and analysis has led to the need for other mono- and multilingual easy-to-use end-to-end systems.

Event coreference resolution (ECR) is only one of the practical use cases that would significantly benefit from being embedded into an end-to-end framework. The goal of ECR is to resolve which textual events refer to the same real-world or fictional event. Consider Examples 1 and 2 below, in which two textual events were taken from two different Dutch (Flemish) newspaper articles:

1. [België neemt het op tegen Kroatië.]$_{Event}$ *EN: Belgium takes on Croatia.*
2. [De laatste groepswedstrijd van de Rode Duivels.]$_{Event}$ wordt maar beter snel vergeten *EN: The last group game of the Red Devils should better be forgotten.*

For human readers, it is immediately apparent that the two events refer to the same real-world occurrence, being the football match between Belgium and Croatia at the 2022 World Cup. In fact, this ability to weave an intricate narrative web, not only within one given text, but also across various texts that are read, is one of our greatest tools for discourse understanding. For computer systems, however, this is much more complicated since these lack the required real-world knowledge. Nonetheless, the ability to embed this information could hugely benefit practical NLP applications, such as template filling [2], automated population of knowledge bases [3], question answering [4] and contradiction detection [5]. While event coreference resolution is a challenging research topic in itself, it inevitably relies on the prior extraction of events; solving both tasks jointly thus seems a logical choice. Within the coreference resolution domain, studies that combine both tasks, either in a joint learning setup or a pipeline approach, are designated as end-to-end [6,7]. In this case, the textual events are first extracted from a collection of raw text, after which coreference is resolved between those events. The ability to immediately extract, analyze and draw links between aspects of document collections is immensely important for present-day commercial applications, such as automatic summarization [8] and content-based news recommendation [9]. In addition, the investigation of end-to-end coreference resolution can be useful, as the relations that are being modeled are inherently discourse-based and often break down both topic and document barriers, whereas most current NLP paradigms are still based on local contextual word-level representations. As is evident from our own understanding of language, local relations are often not enough for a complete understanding of a given text or discourse. Modeling discourse-level semantics can thus be beneficial, not only within the framework of event coreference resolution, but also for text understanding in NLP as a whole.

In this paper, we present our efforts to create the first benchmark end-to-end system for cross-document event coreference for the Dutch language. We first position our research with respect to the state of the art by discussing both the available resources and current methodologies used in the English language domain (Section 2). We then discuss two recently developed corpora for Dutch event detection and coreference, which will jointly be used as training data for the new end-to-end system (Section 3). Subsequently, we discuss the creation of our baseline model for Dutch end-to-end coreference resolution, which consists of a pipeline architecture composed of two state-of-the-art transformer-based methods for event detection and coreference resolution, respectively. In addition, we also train combinations of traditional feature-based algorithms to provide a comparison to our proposed method (Section 4). We conclude this paper by providing a thorough analysis of both pipeline components and by formulating a set of suggestions for further improvements (Section 6).

2. Related Work

2.1. Resources

Data availability is one of the major problems currently plaguing the research domain of event coreference resolution, even for the generally well-resourced English language sphere. Additionally, the corpora that are available typically have their own specific annotation schemas, event topology and general conceptual definition of what exactly constitutes an event. This gravely complicates the applicability of event coreference methodologies on datasets for which they were not originally designed and hinders general comparison. In the section below, we aim to give a brief yet comprehensive overview of publicly available mono- and multilingual event coreference corpora, listing their general setup, applicability and possible flaws.

The OntoNotes corpus [10] has since long been the standard for benchmarking both entity and event coreference resolution systems in the English language domain. This large-scale multi-domain text collection includes Treebank [11], Propbank [12,13] and within-document coreference annotations both at the entity and event level. A notable caveat of the OntoNotes corpus, however, is that no distinction is made between the entity and event labels; as such, they are both simply designated as a mention. While the tasks of entity and event coreference share certain aspects, it should be noted that the challenges researchers are currently faced with for both tasks are largely different. Furthermore, verbal events in particular are not fully defined for the OntoNotes corpus on a fundamental level and can only be designated as a single-word mention if there exists a noun phrase equivalent of the same real-world or fictional event. This in turn makes the data less consistent, especially with the expressed goal in mind to develop practical real-world application driven by coreference resolution algorithms. The ACE corpora, and more specifically ACE 2005 [14], are generally also accepted as a solid evaluation standard. Limited to events of a set of 100 predefined type actions, the ACE 2005 corpus includes event and coreference annotation on a within-document level for both English and, to a more limited extent, Chinese. While the event schema for ACE is fairly limited, its general approach has been adopted and expanded throughout the years. For instance, the TAC-KBP corpora [15] try to build upon the ACE methodology by not only expanding the event typology, but also by including a more complex and information-rich annotation style for the events contained in the documents. This multilingual dataset includes documents for English, Arabic, Chinese and Spanish and considers coreferential links at the within-document level. Note that all of the aforementioned datasets strictly annotate coreferential links on the within-document level and that for cross-document event coreference resolution, available resources are even more scarce. The accepted standard for benchmarking in cross-document research is the monolingual English ECB+ corpus [16], itself an extension of the more limited ECB corpus [17]. The corpus includes a significant number of newspaper documents in which multi-word event spans are annotated using a schema based on the highly informational Rich ERE guidelines for event annotation [18]. The second and final cross-document corpus for English is the more recently developed WEC-Eng dataset [19], which adopts a novel method of semi-automatically leveraging data, where both event mentions and coreference links between events are not restricted to pre-defined topics.

In the Dutch language domain, there exist two event coreference datasets. While the larger of the two constitutes the majority of the data used in our own experiments and will therefore be discussed in detail in Section 3, it is still useful to briefly discuss the more limited MeanTime Newsreader [20] corpus. This richly annotated event dataset contains cross-document level coreference annotations for English, Dutch, Spanish and Italian and deals with largely unrestricted events. However, the documents collected in this dataset are limited to four specific topic domains, and the Dutch, Spanish and Italian sections of the dataset have been machine-translated from the original English texts. Moreover, the event and coreferential annotation of the Italian and Spanish sections have not been carried out manually but rather through cross-lingual projection.

2.2. Methodology

Methods in event coreference resolution usually follow the same paradigms that exist for entity coreference research [21]. Note here that all of the methods discussed below are situated exclusively in the English and Chinese language domains. The very limited research on the Dutch language specifically will be further discussed in Section 4. Currently, three paradigms for resolving coreference are widely used. First and foremost is the mention-pair approach, which transforms the creation of coreferential mention clusters into a pairwise classification task in which pairs of event mentions are generated and a binary classification algorithm determines whether or not these are indeed coreferent. Next, a clustering algorithm is applied to reconstruct an event coreference chain representation of the binary output. The classification algorithms used in these mention-pair setups have followed the natural progression of machine learning methods commonly used in NLP in recent years. In the past, feature-based approaches, such as support vector machines [22], decision trees [23] and deep neural networks [24], were primarily used. From these studies, it became clear that outward lexical similarity was the prime indicator of coreference, with features based on (partial) string matching being most effective [7]. Additionally, features modeling document structure had some limited success in within-document contexts [7,25]. More recently, however, feature-based approaches in coreference resolution have been challenged by transformer-based approaches in which large language models are used to generate strong contextual mention representations, which are then used as a basis for the classification algorithms [26,27]. For the English language domain specifically, span-based transformer approaches [28] have been shown to attain state-of-the-art results [6]. These pre-trained language models are specifically tuned toward encoding longer word sequences, which in turn results in stronger event (multi-word) contextual representations. While mention-pair models perform best overall in coreference resolution tasks, a noticeable shortcoming is their inability to consider an event coreference chain consisting of more than two events collectively. This is because the algorithm boils down to pairwise decisions and not to a decision based on the discourse as a whole. This is exactly what the second paradigm, mention-ranking, attempts to combat to a certain extent. Within this framework, a ranking of a mention's possible antecedents is generated from the feature representation of both the mention and its antecedents. Some algorithms following this approach create coreferential chain partitions of the text as a whole, generating a combined probability of all coreferential relations within it [29]. Finally, a third accepted methodology is that of easy-first modeling. Following the successful application of rule-based multi-pass sieve algorithms in entity coreference studies [30], the multi-stage approach has also been evaluated in event coreference research [31]. Multiple classification rules or *sieves* are set up in order of decreasing precision, resulting in a system where comparatively 'easier' mentions are resolved first. While the rule systems are primarily based on pairwise comparisons between event pairs, there exists the option to embed global coreference cluster information, circumventing, albeit to a minor extent, the aforementioned problems with mention-pair approaches. Additionally, it is possible to further boost the performance of these easy-first methods by including techniques such as within-chain event argument propagation [32] and agglomerative clustering [33].

All methods and algorithms detailed above are strictly for the resolution of coreferential links based on given gold-standard event mentions. However, in an end-to-end setup, those mentions will first need to be extracted from raw text. Generally, two approaches are possible for end-to-end coreference resolution: *pipeline* or *joint* methods. In a pipeline setup, the detection and resolution of event mentions are seen as two distinct components, and any self-contained event detection method can be paired with any of the resolution methods described above. While the implementation of such systems is relatively straightforward and highly modifiable, they notoriously suffer from error propagation, as errors made in one of the components get passed as the input for the next component with no method to reliably rectify them. *Joint* event coreference resolution, on the other hand, attempts to model both the detection and coreference resolution of events at the same time. The joint modeling paradigm can be realized by employing methods such as Markov logic networks [34] and integer linear programming [35] to perform joint inference. The two components in the task can improve one another by embedding background knowledge for each component in the task as a set of constraints. Another approach to joint coreference resolution can be full-on joint learning, where the two tasks are merged as a structured prediction problem and learned through a segment-based decoding algorithm [36]. Overall, joint methods, with joint inference methods in particular, obtain the best results within the field [6], especially when combined with well-performing entity coreference resolution systems [7] and transformer-based architectures [28].

3. Data

Currently, two large event-based datasets exist for the Dutch language: EventDNA (Section 3.1) and ENCORE (Section 3.2). It should be noted, however, that only one of those—ENCORE—includes coreferential annotations. In the following sections, we briefly highlight the main features of these two corpora by discussing their general design and respective annotations.

3.1. EventDNA

The EventDNA corpus [37] consists of a total of 1773 news documents, each of them composed of a title and lead paragraph from a given newspaper article. The article data were sourced from a larger collection of newspaper information collected within the sphere of the NewsDNA project [38]. The NewsDNA collection contains newspaper articles that were published online in between 2018 and 2019 from a large variety of Dutch (Flemish) sources, such as national (*De Morgen*, *Het Nieuwsblad*, *Het Laatste Nieuws*, *De Standaard*) and regional (*Het Belang van Limburg*) newspapers. Additionally, articles published on the news website of the Flemish public broadcasting agency (*VRT News*) were included.

Annotation of the news texts was performed at two levels. First, textual entities were annotated and coreferential links drawn between them at a within-document level. Secondly, textual events were annotated. The event annotation scheme was based on the rich ERE guidelines [18], which provide a framework for the annotation of multi-word event spans with event attributes such as type and subtype. These types are part of a predefined taxonomy, as exemplified for the *Life* event in Table 1 below.

Table 1. Type and subtype for the *Life* events defined in the ERE guidelines.

Event Type	Event Subtype
Life	beBorn
Life	die
Life	divorce
Life	injure
Life	marry

The authors of the corpus also included an *Unknown* type label in order to label important events that do not fall within the predefined ERE taxonomy, meaning that the corpus comprises unrestricted events. Additionally, a set of event characteristics was defined. This includes the importance or prominence (*main event*/*Background event*) of an event within the news document, the tense of the event (*past*, *present* or *future*), negation and sentiment. Finally, each event is composed of a series of *event arguments*. These arguments provide additional information about the real world event and correspond well to the wh-questions: what, who, where, when, why and how? The arguments that can be annotated for each event are dependent on the predefined *event type*. Examples 3 and 4 denote two fully annotated events using the EventDNA annotation schema.

3. [[Moordenaar Kitty Van Nieuwenhuyse]$_{PERSON}$ komt niet vrij.]$_{Event|Justice|ReleaseParole}$ *EN: Kitty Van Nieuwenhuyse's murderer will not be released.*
4. [De finale van het WK voetbal 2022.]$_{Event|Unknown}$ *EN: The final of the 2022 World Cup.*

3.2. ENCORE

The ENCORE corpus [39] is the largest cross-document unrestricted event coreference corpus in the Dutch language domain. It contains 1015 full newspaper documents, sourced from the aforementioned NewsDNA data collection [38]. However, it should be noted that none of the documents in the ENCORE corpus are present in the EventDNA dataset. Like many event-based corpora, entity-level annotations are also included, but only when said entities are part of an annotated event span and coreference relations between those entities are also indicated on a within-document level.

Unlike the EventDNA corpus, the textual event annotations within the ENCORE corpus are based on different annotation guidelines. Within ENCORE, event annotations were inspired by the ECB+ event coreference corpus [40] in order to provide a more streamlined comparison between English and Dutch event coreference resolution. This implies that within ENCORE, events can be both nominal and verbal, single- or multi-word expressions and typically consist of a central event action (or trigger) and a series of spatio-temporal and participant arguments. A noticeable difference between the ENCORE corpus and EventDNA is the lack of a predefined event taxonomy, allowing for unrestricted event annotation in the dataset. The absence of an event taxonomy naturally results in generic arguments, such as *location*, *time*, *human participant* and *non-human participant*, rather than event-specific arguments, which are found in many of the earlier-mentioned corpora.

Examples 5 and 6 illustrate how the events in Examples 3 and 4 have been annotated within the ENCORE framework.

5. [[Moordenaar Kitty Van Nieuwenhuyse.]$_{HUMANPARTICIPANT}$ komt niet vrij$_{Action}$]$_{Event}$ *EN: Kitty Van Nieuwenhuyse's murderer will not be released.*
6. [De [finale]$_{Action}$ van het WK voetbal 2022$_{TIME}$.]$_{Event}$ *EN: The final of the 2022 World Cup.*

In addition to the actual events, a set of event characteristics, such as prominence, realis and sentiment were also annotated, similarly to the events in the EventDNA corpus. In the ENCORE corpus, a total of 15,407 events were annotated, of which 3698 were nominal constituents and 11,709 were verbal events. Additionally, a total of 35,315 event arguments were also annotated in the entire corpus. Table 2 lists the number of document, tokens and events annotated in both Dutch corpora.

Table 2. Respective sizes of the two corpora, based on the number of documents, tokens and annotated events.

	EventDNA	**ENCORE**
Documents	1773	1015
Tokens	106,106	849,555
Events	7409	15,407

Arguably, the most important component in the ENCORE corpus is the annotation of coreference, which is annotated at both the entity and event levels. While entity coreference annotations are restricted to the within-document level, event coreference annotations are indicated both within and beyond document borders. The latter cross-document annotations were performed within larger topic clusters containing documents relating to the same events. For two events to be considered coreferent, three conditions had to be fulfilled: the two events must happen at the same time (1), in the same place (2), and the same participants must be involved (3). In total, the corpus contains 1018 within-document and 1578 cross-document event coreference chains (i.e., clusters of 2 or more coreferring events), making it comparable in size to the large-scale English-language datasets that were discussed in Section 2. Furthermore, each chain contains, on average, seven events and the cross-document chains span an average of 5–6 documents.

4. Experiments and Methodology

In our experiments, we aim to perform end-to-end cross-document event coreference resolution for the Dutch language for the first time. Concretely, we aim to extract events from raw text and predict which of those events refer to the same real-world event.

Our proposed model comes in the form of a pipeline architecture, where the event detection component feeds directly into the event coreference classification algorithm. To this purpose, we merge two fully closed pre-trained models for detection [41] and coreference resolution [25], respectively. Figure 1 shows a schematic representation of the proposed end-to-end pipeline.

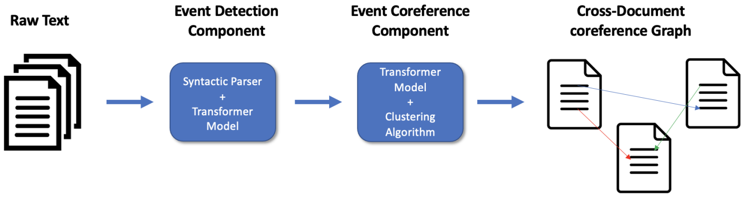
Raw Text
Event Detection Component
Event Coreference Component
Cross-Document coreference Graph
Syntactic Parser + Transformer Model
Transformer Model + Clustering Algorithm

Figure 1. Schematic representation of the pipeline model.

As discussed above in Section 2, joint models often have an edge over pipeline architectures; however, we still opt for the latter as our preferred setup for two reasons. First, the lack of existing work on Dutch end-to-end event coreference resolution obliges us to build a method from scratch. While an existing architecture from another language domain, such as English, could be used here, it has been shown in previous work on Dutch gold-standard event coreference resolution that extending these models to incorporate Dutch typically leads to a significant drop in performance [25]. Moreover, the aforementioned different styles of event and coreference annotation and corpus structure, both between various English-language datasets and between the most commonly used benchmark datasets and the Dutch corpora used in this paper, would gravely complicate both the adaptation and comparison of existing systems. The second reason to opt for a pipeline architecture is more fundamental. The main goal of this paper is to tackle the task of Dutch end-to-end event coreference resolution for the first time and as such create a benchmark. We believe that combining existing components is the most logical choice for such a baseline. Additionally, by separating the steps of the task we allow ourselves to perform more fine-grained error analyses on these individual components, which in turn can lead to the development of better performing and even end-to-end joint models in the future.

In the sections below, we discuss the individual components of the proposed pipeline, our experimental setup, and the various metrics for evaluating coreference resolution.

4.1. Event Extraction Model

In order to detect events from raw text, we employ the Dutch extraction model that was presented in Desot et al. [41], which was trained and tested on the EventDNA corpus [37]. This method for multi-word event extraction consists of a two-step approach, which first selects candidate event spans based on syntactic parsing and then applies a transformer-based classification algorithm in order to assign the candidate span to one of three categories: *no event*, *main event* or *background event*. In addition, we compare this approach to a baseline token-level conditional random field (CRF) classification algorithm [37] in order to compare the proposed algorithm to more traditional feature-based machine learning approaches to event detection.

4.1.1. Selection of Candidate Events

In order to select candidate multi-word event mentions, raw text is first parsed using the Dutch syntactic parser Alpino [42]. Then, a series of manually constructed rules are applied to the parser output in order to obtain main and subclauses, which are subsequently encoded and fed into the classification algorithm.

Candidate event extraction through syntactic clauses has two notable caveats that need to be addressed before discussing the second component of this algorithm.

First and foremost, because this extraction algorithm only considers main and subclauses in the sentences as possible events, nominal events embedded within another event are typically not considered. This inevitably means that some events will not be detected, as illustrated in Example 7.

7. De wapenstilstand van 11 November maakte een einde aan de Eerste wereldoorlog.
   EN: The truce on the eleventh of November ended the first world war.

In this case, only the clause *De wapenstilstand van 11 November maakte een einde aan de Eerste wereldoorlog* will be extracted, but the nominal constituent events *De wapenstilstand* and *de Eerste wereldoorlog*, both events in their own right, will not be considered.

A second and more general issue is that the extracted clauses do not necessarily fully overlap with the annotated gold-standard events. In order to allow for evaluation of the coreference component in the pipeline, a mapping needs to be created where extracted spans are mapped as accurately as possible to their gold-standard equivalent. Naturally, extracted spans that are not part of an annotated event lack such mapping, but are still considered in the coreference resolution evaluation. Manually mapping candidate spans to gold-standard events would be both intensive and time-consuming. We therefore opted, following Desot et al. [41], for a transformer-based mapping algorithm, where span embeddings are generated through a SentenceBERT [43] model. Then, cosine similarity is computed per sentence between all extracted clauses and annotated gold-standard events in that sentence. The best-matching extracted clause is then mapped to the respective gold-standard event. From that moment on, the candidate clause and annotated event share a unique internal ID, which is subsequently used to compute the evaluation metrics for each of the tasks in the event detection approach. Note that the event mapping is only used for the calculation of the various evaluation metrics and not in the classification of candidate events itself. Figure 2 below provides a detailed visualization of this mapping process.

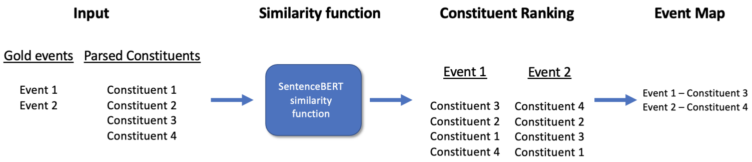

Figure 2. Visualization of the mapping of events and candidates in the same sentence.

4.1.2. Classification of Candidate Events

After extracting and mapping potential multi-word event spans, these are subsequently classified as either an event or no event through a transformer-based classification algorithm. Standard transformer models consist of an encoder–decoder network employing both self-attention (encoder) and attention (decoder) and have been shown to perform extremely well in various domains, ranging from vision to NLP [44]. BERT (bidirectional encoder representations from transformers) models [45] are able to create strong contextual vector representations of language by stacking several encoder layers (12 in a regular-sized setup) and enhancing these by applying multi-head self-attention. These models are first trained to generate strong contextual embeddings through masked language modeling (MLM) and next sentence prediction (NSP) tasks on large quantities of unlabeled data (pre-training). Then, pre-training model weights are initialized, and the models are trained again on a smaller labeled dataset, allowing them to generate task-specific contextual language embeddings (fine-tuning). Since their inception, BERT-based models have shown to attain state-of-the-art results on a large variety of NLP tasks, such as emotion detection [46], relation extraction [47] and question answering [48].

In the setup presented here, we rely on the Dutch monolingual BERTje model [49]. BERTje was pre-trained on a total of around 2.4B tokens of high-quality Dutch texts, which include the Dutch Sonar-500 [50] and TwNC [51] corpora, Wikipedia data, historical fiction and a large collection of Dutch online newspaper articles collected over a 4-year period. As a significant portion of the BERTje pre-training data is made out of newspaper articles, we believe this model is particularly fit for event-related tasks.

Concretely, we employ the model used by Desot et al. [41] to classify the candidate event spans. This model was fine-tuned on a part of the EventDNA dataset. As this is a span-based classification task, a special *[CLS]* classification token was inserted at the beginning of each candidate span, serving as aggregate representation for the entire span. Each span was then encoded individually by the BERTje model and classified through a softmax-activated classification layer. Note that in the original paper, candidate event spans were judged to be one of three predetermined labels: *main event*, *background event*, or *no event*. Since we wish to directly incorporate the output of the event detection step in our pipeline end-to-end setup, we forgo the aforementioned distinction between the labels of main and background events and simply aggregate the mentions predicted as *main event* and *background event* under the common label of *event*. The reason for this change is that while the difference between main and background events can be useful for integration in event-based applications, such as topic detection [52,53] and story segmentation [54], this distinction serves no real purpose in the context of cross-document event coreference resolution, where main events in one document (8) can be background events in another document (9) and vice versa.

8. [vakbond legt het werk neer op 21 December.]$_{Event-Main}$ *EN: Union will strike on 21 December.*
9. [De nieuwe staking]$_{Event-Main}$ doet denken aan [de vakbonsacties van December vorig jaar]$_{Event-Background}$ *EN: The new strike is reminiscent of the actions in December last year.*

4.2. Event Coreference Model

As detailed in Section 2, two widely used paradigms exist for resolving coreference between entities or events: mention-pair and mention-ranking approaches. We opt to exclusively work with a mention-pair approach in this paper for two reasons. First, it has been shown in previous work that in event settings, for both English and Dutch, mention-ranking methods are generally less effective [6,25]. Second, mention-ranking is somewhat problematic in the context of cross-document coreference resolution, both from a conceptual and practical point of view. In order to compute a ranking of antecedent mentions, one needs to establish what the antecedents of a candidate mention are in the first place. This is impractical in a cross-document setting, as there is no set order for events in a collection of different documents, and hence no consistent way of generating the antecedent list for a candidate event mention. While some mention-ranking approaches can partly mitigate this by constructing a ranking of document partition vectors instead [29], it has been shown in previous work that extrapolating these models to a cross-document setting can pose serious scalability issues [25].

The mention-pair algorithm we propose to use as our second step in the end-to-end event coreference resolution pipeline is conceptually not different from the methods discussed in Section 2. For each pair of extracted event mentions, a pairwise score is computed to determine the absence or presence of a coreferential link. We select the MP BERTje model discussed in De Langhe et al. [25] as our coreference resolution algorithm of choice. In order to obtain pairwise scores for our extracted event mentions, we first encode each possible intra-cluster (see Section 3) event pair in the ENCORE dataset. Event mention pairs are concatenated, tokenized and fed to the BERTje encoder. A special *[SEP]* token is inserted between the two event mentions to indicate where one ends and the other begins. Similar to the event detection task, we use the token representation of the classification token *[CLS]* as the aggregate embedding of each event pair, which is subsequently passed to a softmax-activated classification function. Finally, the results of the binary text pair classification are passed through a clustering algorithm in order to obtain output in the form of coreference chains.

Similar to the event detection component, we also experiment with a feature-based coreference resolution baseline in the form of a gradient-boosted tree algorithm popularly known as XGBoost [55]. For this approach, we use a collection of lexical similarity features, such as event span and action similarity based on cosine distance. Additionally, we use certain discourse and constraining features that have been applied for both English [7] and Dutch [25] ECR studies. These include time and location matching for the events pairs, coreferential information on the events' participants and potential matches for meta-linguistic event traits, such as their realis and sentiment.

4.3. Experimental Setup and Evaluation

In accordance with existing literature on end-to-end event coreference resolution [6,7], we evaluate our system directly through the performance of the coreference resolution component, rather than evaluating each aspect of the pipeline separately. Before presenting our experimental results, however, two notable aspects of our evaluation strategy are addressed below.

4.3.1. Data and Upper Bound

As described in Section 3, we have access to two Dutch large-scale event corpora (Table 2). Though both corpora comprise event annotations, we train the event detection component using the same EventDNA data configuration, random seed and hyperparameter values used in the original paper by Desot et al. [41]. Since no data for Dutch cross-document event coreference resolution are available other than the ENCORE corpus, we retrain the MP BERTje coreference model presented in De Langhe et al. [25] using again an identical setup with respect to splitting the data and hyperparameter optimization. The evaluation of the entire end-to-end system is exclusively performed on the original held-out test set of the ENCORE corpus. Table 3 below provides a schematic representation of the data size configurations used in the respective components and highlights training and testing portions of the dataset that are actively used in the end-to-end pipeline. Logically, as hyperparameter configurations were drawn directly from the respective papers for both components, the development set of ENCORE corpus remained unused for this experiment.

Table 3. Number of events in the train, development and test splits for the EventDNA and ENCORE corpora. Parts of the training set that are actively used for the training and evaluation of the end-to-end setup are highlighted in bold.

	Training	Development	Test
EventDNA	**5927**	741	741
ENCORE	**10,785**	2312	**23,12**

A major benefit of training the pipeline components identically to their original implementation is that there is an upper bound to the performance of the end-to-end algorithm, being the performance of the MP BERTje coreference model with gold standard events. End-to-end coreference systems are usually only evaluated through the result of the coreference component, while the event detection task is seen as secondary. The upper bound presents us with a unique opportunity to evaluate the event detection component in a more direct fashion, as we have means to directly gauge the impact of certain modifications in the detection component with respect to the overall upper bound. From an analysis point of view, this can, in turn, provide valuable insights for improving the event detection component in future research.

4.3.2. Metrics

Before discussing the results of our end-to-end experiment, it is useful to mention that coreference resolution is typically evaluated through a series of cluster or link-based metrics. The choice of which method to use is a point of contention among researchers investigating coreference resolution, and many different evaluation metrics have been proposed throughout the years. Usually, an average F1 is computed from three different metrics (MUC, B3, and CEAF), often referred to as the unified CONLL F1 score [56]. In the sections below, we briefly discuss the advantages and drawbacks for each of these scores. Additionally, we also make a case for the inclusion of a more recently proposed evaluation metric within the CONLL framework.

MUC

The MUC evaluation metric [57] is one of the earliest proposed methods for reliably evaluating coreference. It is inherently based on the number of missing coreferential links that need to be inserted in the response clusters in order to match the gold-standard coreference clusters. A major benefit of the MUC metric is its robustness with respect to singleton clusters (i.e., mentions that do not have any coreferential links), as singleton clusters are known to artificially inflate scores when present. While singleton distortion is kept to a minimum, the metric does not discriminate between different types of errors and more severe errors are punished as hard as smaller errors [58]. This, in turn, sometimes results in scores which do not fully correspond to human judgment and are more difficult to interpret.

B3

In contrast to the MUC metric, B3 precision and recall are calculated based on correct entities/events in the response clusters. While the B3 metric addresses several inconsistencies introduced by MUC, it also suffers from singleton distortion due to the entity-based calculation. Additionally, Pradhan et al. [59] posit that repeated gold entities or events will boost the scores artificially and subsequently show that this results directly in problematic cases when dealing with parse trees where an NP node has a single, pronominal, child. In the latter case, both nodes (i.e., child and NP node) are resolved and result in an inflated score.

CEAF

The CEAF coreference evaluation metric [60] is based on finding an optimal alignment between entity/event chains in the system output and corresponding gold-standard chains. Concretely, the Kuhn–Munkres algorithm is used to align chains of coreferring entities in key and response based on a given similarity metric. Being an entity-based evaluation, it suffers from singleton distortion. Even more so than the aforementioned B3 metric, as in the calculation of the CEAF metric, the total size of the aligned coreference chains is not taken into account, meaning that a correctly predicted singleton cluster contributes as much to the final score as a cluster with a large number of entities [61].

LEA

More recently, the link-based entity-aware (LEA) metric was proposed by Moosavi and Strube [58]. Learning from the shortcomings of the previously discussed metrics, LEA computes a relative importance for each of the coreferential chains in the output based on the size of the respective chain. Furthermore, each of the response chains is checked for a minimal partial overlap (>1) with the gold-standard chains, reducing the influence of singletons on the overall score. It has to be noted here that recent studies suggest that singleton mentions can still have an effect, although the distortion is significantly smaller than for other metrics [62]. We believe that the improvements introduced in the LEA metric are significant and can help in a more thorough understanding and better interpretation of research in coreference resolution. We, therefore, will include LEA scores in our own evaluation of the end-to-end pipeline system.

5. Results

In Table 4, we present the results when experimenting with different feature-based and transformer-based combinations of both components. In general, scores are slightly worse, but still comparable, to those obtained in English-language coreference studies (0.42 CONLL F1 and 0.44 CONLL F1 for span-based transformers in pipeline and joint architectures, respectively) [6]. As expected, the transformer-based pipeline (bottow row) significantly outperforms the feature-based baselines for both the CONLL F1 and LEA F1 scores. As is often reported in literature, the LEA F1 scores are generally lower compared to the CONLL score. If we compare the upper bound (0.59 CONLL F1 and 0.39 LEA) using gold-standard events with the best score obtained by the transformer pipeline (0.27 CONLL F1 and 0.08 LEA), there is still considerable room for improvement in the event-detection component. Nonetheless, the transformer pipeline architecture results in a solid baseline reference for future work.

Table 4. Results for the end-to-end pipeline experiment.

Detection Component	Coreference Component	CONLL F1	LEA F1
CRF	*XGBoost*	0.17	0.04
CRF	*BERTje*	0.21	0.05
BERTje	*XGBoost*	0.22	0.06
BERTje	*BERTje*	**0.27**	**0.08**

In the next two sections of this paper, we will present an in-depth analysis of both components.

6. Analysis and Discussion

The goal of this section is to thoroughly analyze each component in the end-to-end pipeline. Previous studies in event coreference resolution exclusively focus on architectural and algorithmic optimization, disregarding detailed error analysis. Nonetheless, we believe that by analyzing the output of the systems discussed, we can learn valuable information regarding the errors made in the classification process. Additionally, finding error patterns in the pipeline can be immensely valuable with respect to future research endeavors.

6.1. Event Detection Component: The Effect of Event Extraction Filtering

In general, we found that the event detection component extracts a significantly larger number of events (10211) than actually present in the test data (2312). This is due to the inherent structure and annotation style of the EventDNA corpus on which the event detection component was trained. As stated in Section 3, the EventDNA corpus is composed of headers and lead paragraphs from newspaper articles. The ENCORE corpus, on the other hand, comprises full newspaper articles. Previous analyses of both corpora revealed that important events are mainly found in the first three sentences of a newspaper article and at the start of newspaper articles [39,63]. This means that the EventDNA corpus as a whole is much more dense and rich in events compared to the ENCORE corpus from which the held-out test set was derived for our experiments. This is also supported by the comparison given in Table 2. The most trivial explanation for the overgeneration of events is thus a mismatch between training and testing data in the event detection component. A logical conclusion would be to state that the EventDNA corpus might not be suited for full-text newspaper analysis, but should rather be used for the development of applications which aim to model short and dense texts. In addition, we believe there is another reason which makes the two corpora less comparable than initially thought. On a fundamental level, events in coreference studies are usually defined as real-world events with a clear timespan, location and participants [16]. These informative characteristics are crucial, not only from a classification point of view, but also with respect to the applications in which event coreference can be used, such as content-based news recommendation. Events in detection studies on the other hand tend to only require an action in the form of a verb to be considered as an event [64]. This is particularly noticeable when examining *speech* events. These constitute a class of events which are typically less semantically rich, such as in Example 10.

10. Fouad Belkacem zegt dat hij zich zal verzetten tegen de uitspraak.
EN: Fouad Belkacem says he will resist the verdict.

While both *zeggen/say*) and *verzetten/resist* can be considered as an event here, only the latter is usually considered in coreference studies. While *speech* events can be useful in certain downstream tasks, such as timeline generation [65,66] or automatic summarization [67], these events hold very little informational value on their own, making them less optimal both from a data and practical perspective [39]. While *speech* events are annotated in the EventDNA corpus, contributing further to its event density, these events were omitted from the ENCORE corpus. Note, however, that for the ENCORE corpus, events of this type can occasionally be annotated when they constitute important happenings within the context of the document. This includes, among others, events such as courtroom verdicts and speeches.

In order to overcome this overgeneration of events, we gradually applied a set of rudimentary pruning heuristics to the collection of predicted events for the best CRF and transformer-based pipeline. This includes the removal of single-token events, events that are predicted with the *background* label and the removal of *speech* events. Table 5 displays the effects of the various pruning steps on the number of predicted events and the overall performance of the pipeline model, expressed in CONLL F1 and LEA. In order to gain insight into how many of the actual gold-standard events are found by the event detection component, we also included the subset of mapped gold events (bottom row).

Table 5. Effects of pruning heuristics showing the influence on the CONLL and LEA scores as well as the total number of events extracted.

	CRF			BERTje		
	CONLL	**LEA**	**Events**	**CONLL**	**LEA**	**Events**
No_Pruning	0.22	0.05	11089	0.27	0.08	10211
Remove single-token events	0.26	0.07	9581	0.29	0.10	9038
Remove speech events	0.28	0.07	7436	0.32	0.13	7015
Remove background events	0.31	0.13	5722	0.34	0.15	5813
Remove background + speech	0.35	0.20	5008	0.39	0.24	4487
Gold mapped events	0.38	0.22	1968	0.46	0.27	2003

We should note here that if we only consider predicted events that have a mapped link to a gold-standard event, the overall CONLL and LEA scores move significantly closer to the established upper bound score of 0.59 and 0.39, respectively. This might imply that while many of the gold-standard events are indeed found, the noise created by the overgeneration of events is the main issue in the detection module's performance. Simply excluding events predicted with the *background* label removes a significant number of events from the pool of predicted events, while providing an increase in the global CONLL and LEA scores. This might imply that many of the parsed constituents were simply assigned the *background* label, where there should have been no event label at all. Another interesting observation is the noticeable impact of excluding single token events. This is somewhat unexpected, as the Alpino parser output should correspond to entire main and subclauses, not single words. This might point to more fundamental errors with the syntactic parsing of Alpino itself, which are then regrettably propagated through the end-to-end pipeline. Finally, we can also infer that our initial suspicions regarding the inclusion of *speech* events were correct, as removing verbal events from the predicted events pool also has a positive effect on the classification scores. While it is self-evident that correcting a mismatch between training and testing data results in a better overall result, this does, however, open up an interesting discussion with respect to the interaction between event detection and event coreference studies. The plethora of annotation schemes, datasets and various definitions of what constitutes events in the first place greatly hinders the development of the event NLP domain as a whole. Even within the subfield of event coreference studies, existing datasets are hardly comparable [7]. This in turn creates an environment where benchmarking in general becomes problematic, either due to the technical effort required to make existing models compatible with different datasets, or due to the vast conceptual differences between datasets themselves. While beyond the scope of this paper, these observations do raise the question of whether or not there is a need for an organized effort in standardizing certain conceptual principles with respect to the event domain as a whole.

6.2. Event Coreference Component: The Importance of Lexical Similarity in Transformer Models

As stated before, a thorough analysis of the coreference component in the pipeline is hard. The deep neural architecture of the transformer model prohibits a detailed feature analysis, and the interpretation of the classification decision is near impossible. Nonetheless, there are some interesting aspects of this component which can be addressed in closer detail. Recent work on event coreference resolution using transformer architectures has suggested that the applicability of these models is limited, as all classification decisions are based on outward lexical similarity [68]. Nonetheless, it is generally accepted that transformer architectures are able to to embed much more information at both the syntactic and discourse levels of the text [69]. Concretely, this implies that in a mention-pair approach, an event pair with high lexical similarity between the two mentions will always be classified as positive, while pairs with a comparatively lower degree of lexical similarity will almost always be classified as being non-coreferent. This is problematic for two reasons. First, while the degree of lexical similarity can be an important indicator of coreference, as suggested by the literature [7], there are still exceptions where the opposite is true, such as in Example 11, where a high degree of lexical or contextual similarity does not necessarily result in a coreferential link. Example 12 denotes the comparatively rarer situation where two mentions that are not lexically similar do refer to the same real-world event. If we calculate the cosine similarity of the two mention pairs each time, we obtain degrees of similarity of 0.851 and 0.542 for Examples 11 and 12, respectively.

11. (a) De Franse president Macron ontmoette de Amerikaanse president voor de eerste keer vandaag. *EN: The French president Macron met with the American president for the first time today.*
    (b) Frans President Sarkozy ontmoette de Amerikaanse president. *EN: French President Sarkozy met the American president.*
12. (a) De partij van Amerikaans president Trump verloor zwaar vandaag. *EN: American president Trump's party lost heavily today.*
    (b) Trump lacht groen bij de midterms. *EN: Sour laughter for Trump at the midterms.*

A second reason is that there are many mention pairs with only an average degree of outward similarity, which a model would then never be able to correctly classify. If this were indeed the case, a significant portion of coreferring mentions in the dataset cannot be resolved by current means. This in turn could mean that other paradigms need to be explored in order to push performance in order to create practical event coreference resolution systems.

In this section, we propose to test this hypothesis by providing a two-fold analysis of the outward lexical similarity between event pairs in the coreference classification algorithm. For each event pair, we embed the mentions using a simple word2vec [70] algorithm and calculate the cosine distance between them. We then plot a graph in which we calculate the average classification accuracy between mentions in a series of cosine similarity intervals. The second part of our analysis consists in visualizing the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) cases in our evaluation set by plotting the average cosine distance for each group.

As can be inferred from Figure 3b above, the average cosine distance between mention pairs is highest for the true positive cases (green bar) and lowest for the true negative cases (red bar). If we consider the erroneously classified mention pairs, the average cosine distance is still relatively high for the false positive cases (blue bar), but rather low for the false negative pairs (orange bar). All of this seems to confirm that also in a transfer setup, the classification decision for coreference is mostly based on outward lexical similarity rather than on deeper textual discourse features, the latter being the actual underlying principles governing coreferential links. Simply put, mentions are classified as coreferent because they are lexically similar and designated as non-coreferent if they are not. The bulk of errors are then made on text pairs that are either lexically similar, but not coreferent (Example 11), or not outwardly similar, but coreferent (Example 12). This is also supported by the graph in Figure 3a, which shows that the average classification accuracy is highest in the low-similarity intervals. As our dataset is heavily skewed toward negative examples, the average classification accuracy is high for non-similar pairs, where the bulk of negative examples are located. However, the classification accuracy drops significantly for more similar text pairs, indicating that many of the cases, such as Example 11, are erroneously classified as positive.

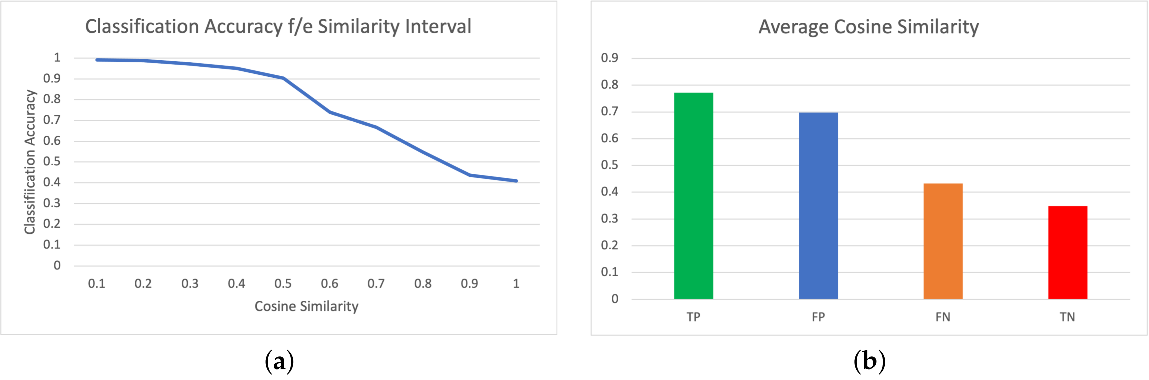

Figure 3. Lexical similarity analysis of the transformer model's output. (**a**) Average accuracy per similarity interval. (**b**) Average cosine distances for the best-performing model.

We can further support this argument by plotting the interval graphs for both the average precision (AP) of positive (coreferent) and negative (non-coreferent) mention pairs. In both graphs, we calculate for each interval the number of correctly classified pairs for each class divided by the total number of pairs in that class for that specific interval.

Figure 4 clearly illustrates that the AP for the positive class greatly increases when similarity is high, while it is lower for non-similar mention pairs, even though these are actually coreferent. We see a similar, but inverse, trend for the negative class. As a whole, this might indicate that the best method proposed in this paper (a transformer-based approach following the mention-pair paradigm) is insufficiently equipped to deal with more complex coreferential classification decisions. This is in line with earlier research which has shown that also feature-based methods suffer from this fundamental flaw, even when structural discourse and argument features are included [7]. If the problem is indeed inherent to the mention-pair paradigm in general, a general change in approach to coreference resolution should be taken in future research in order to alleviate the issue and create coreference algorithms that are well-equipped to deal with all aspects that define coreferential relations and not just lexical properties.

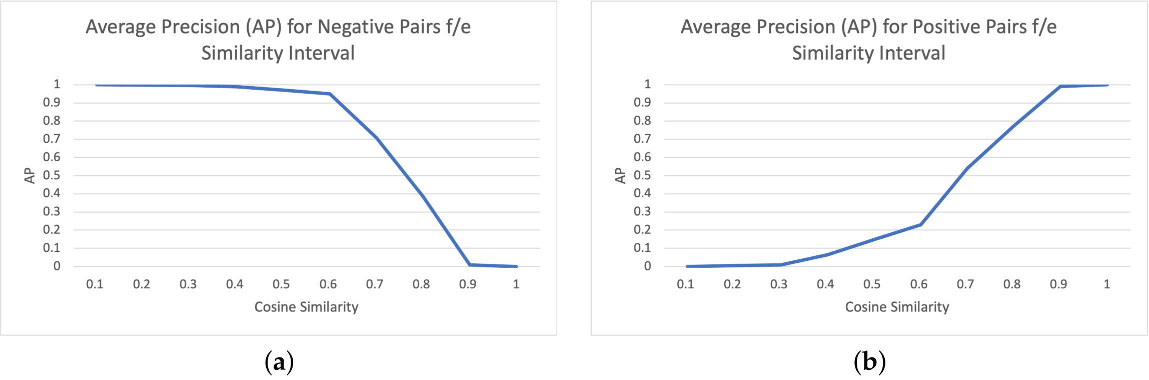

Figure 4. Average precision per cosine similarity interval for positive and negative pairs. (**a**) Average precision for negative pairs. (**b**) Average precision for positive pairs.

7. Ablation Studies

Based on the above analyses, we can conclude that there are two major problems with the end-to-end pipeline in its current form. While the development of a new coreference resolution paradigm is beyond the scope of this paper, the first problem, event overgeneration, can be addressed.

We propose to retrain the entire event detection component by augmenting the original EventDNA data with the events compiled for the ENCORE corpus. So as to not indirectly contaminate the detection component with the test set used for evaluating the end-to-end pipeline, we remove those documents that were used in the held-out test set. By merging the two datasets, we hope to unify the two event data corpora in the Dutch language domain and to strike a balance between the two conceptual approaches that were used when compiling and annotating them. Table 6 compares the effect of retraining the event detection component using various training data configurations of both corpora. We retrain both the baseline token-based CRF classifier and the BERTje transformer classification algorithm. We also add the baseline scores obtained in Section 5 for easy comparison. Like before, the pipeline is entirely evaluated through the downstream event coreference task.

Table 6. Overview of training the models with different data configurations.

Model and Configuration	CONLL	LEA
CRF$_{EventDNA}$	0.22	0.05
BERTje$_{EventDNA}$	0.27	0.08
CRF$_{ENCORE}$	0.29	0.10
BERTje$_{ENCORE}$	0.34	0.16
CRF$_{EventDNA+ENCORE}$	0.28	0.08
BERTje$_{EventDNA+ENCORE}$	**0.37**	**0.18**

As expected, retraining the event detection algorithm with the (more consistent) ENCORE data provides us with better a better eventual coreference classification score. It should also be noted that when augmenting the original EventDNA data with the ENCORE corpus, results are even better. This suggests that corpora with different conceptual annotation schemes may still be compatible to a degree and that the merging of various event corpora in the same language, but potentially also across languages, may prove useful in boosting event coreference resolution performance.

Finally, we apply the same pruning heuristics as in Section 6.1 in order to obtain a better idea of how many of the test set events are detected in the entire pool of predicted events. Table 7 shows the applied filters for each of the retrained BERTje models.

Table 7. Pruning results for the newly trained models.

	ENCORE			EventDNA + ENCORE		
	CONLL	**LEA**	**# Events**	**CONLL**	**LEA**	**# Events**
No_Pruning	0.34	0.16	6481	0.37	0.18	5326
Remove speech events	0.36	0.18	5421	0.41	0.20	4750
Remove background events	0.40	0.21	4809	0.43	0.22	4750
Remove background + Speech	0.44	0.25	4677	0.46	0.26	4159
Gold mapped events	0.52	0.32	2061	0.54	0.36	2183

As can be inferred from Table 7, when only considering the mapped gold-standard events (bottom row), the coreference resolution score moves closer to the upper bound (0.54 versus 0.57 CONLL F1), indicating that most of the gold-standard events in the test set are found. We can also infer that the overgeneration of events remains, albeit to a lesser degree given that almost half of the superfluous events are not detected when combining the two corpora—a persistent problem in the event detection component of the pipeline. However, the fact that most gold-standard events in the test data are found can indicate that with solid and motivated pruning heuristics, the component can be further optimized in the future.

8. Conclusions

In this paper, we reported on the creation of the first end-to-end cross-document event coreference resolution system for the Dutch language, and set a baseline score, which can be used as a point of reference in future research. We first gave a broad overview of existing resources and methodologies in English event coreference studies. We then merged two prominent algorithms for the respective detection and coreference resolution of textual events.

In our initial experiment, we trained the event detection component on the Dutch EventDNA corpus and used the ENCORE coreference corpus for training the coreference resolution component. Additionally, we set aside part of the ENCORE corpus for the evaluation of the entire end-to-end setup. Results were consistent with findings for English language studies. Both the CONLL (0.27) and LEA (0.08) evaluation scores were understandably lower than the respective upper bound scores (0.59 and 0.39), obtained by training and testing the event coreference component on gold-standard event data.

We then conducted a thorough analysis of both pipeline components in order to identify the type of errors that were made in the event detection and event coreference tasks, respectively. The insights gathered from this error analysis can be used to improve future research within the field. For the event detection component, we found that most gold-standard events in the test data were found, but that there was also a significant overgeneration of candidate event mentions. We hypothesized that even though the EventDNA and ENCORE corpora share many similarities and that their respective annotations do not differ immensely, the annotation density of the EventDNA data was in part responsible for this. We tested our hypothesis by gradually pruning the output of the event detection component, based on our knowledge of the respective annotation strategies, and found that with logically motivated pruning heuristics, we could drastically improve the output scores of the end-to-end system as a whole. Additionally, in Section 7, we found that retraining the event detection algorithm using the EventDNA and a previously unused part of the ENCORE data also positively influenced the result. Nonetheless, it is apparent that merging event detection and event coreference corpora for the creation of end-to-end systems should be done with an eye for the respective methodologies and philosophy with which they were designed, especially given the many methodologies in existence, as extensively described in Sections 2 and 3. The event coreference resolution component in the pipeline was investigated by performing a more fine-grained analysis of the individual mention pairs. Previous research has suggested that much like the feature-based methods of the past, transformer-based mention-pair classification is primarily routed in simple lexical similarity, even though BERT-based models are known to be able to model certain syntax and discourse features in other tasks. By calculating cosine similarity between various mention pairs, we found that lexical similarity is indeed the prime feature used in transformer-based coreference classification. While the mention-pair paradigm presently still obtains state-of-the-art scores in many coreference-based tasks, the observations in Section 6 might indicate that in the future, certain cases of coreferring mentions (i.e., those with high similarity but no coreference and those with low similarity but coreference) will remain highly problematic. It is possible that this can be mitigated by going back to mention-ranking and easy-first paradigms used in the past, as the ability to model cluster and wider discourse information between coreferent mentions can be a valuable asset in this regard.

In future research, we will devote ourselves to further improving the benchmark score presented in this paper and improve upon the various components in the end-to-end pipeline. Additionally, we will follow the lead of English-language research in event coreference resolution and investigate the possibility of extrapolating the various joint learning and joint inference approaches to the Dutch language domain.

Author Contributions: Conceptualization, L.D.L., T.D., O.D.C. and V.H.; methodology, L.D.L., T.D., O.D.C. and V.H.; software, L.D.L. and T.D.; validation, L.D.L.; formal analysis, L.D.L.; investigation, L.D.L.; resources, L.D.L., T.D., O.D.C. and V.H.; data curation, L.D.L., T.D., O.D.C. and V.H.; writing—original draft preparation, L.D.L., O.D.C. and V.H.; writing—review and editing, L.D.L., O.D.C. and V.H.; visualization, L.D.L.; supervision, O.D.C. and V.H.; project administration, O.D.C. and V.H.; funding acquisition, O.D.C. and V.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is part of the ENCORE project which is funded by the Research Foundation–Flanders, Project No. G013820N.

Data Availability Statement: Data available on request due to restrictions, e.g., privacy or ethical. The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the data are not being made publicly available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kang, Y.; Cai, Z.; Tan, C.W.; Huang, Q.; Liu, H. Natural language processing (NLP) in management research: A literature review. *J. Manag. Anal.* **2020**, *7*, 139–172. [CrossRef]
2. Humphreys, K.; Gaizauskas, R.; Azzam, S. Event coreference for information extraction. In Proceedings of the ACL/EACL Workshop on Operational Factors in Practical, Robus Anaphora Resolution for Unrestricted Texts, Madrid, Spain, 11 July 1997; pp. 75–81.
3. Ji, H.; Grishman, R. Knowledge base population: Successful approaches and challenges. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 1148–1158.
4. Narayanan, S.; Harabagiu, S. *Question Answering Based on Semantic Structures*; Technical Report; International Computer Science Inst: Berkeley, CA, USA, 2004.
5. De Marneffe, M.C.; Rafferty, A.N.; Manning, C.D. Finding contradictions in text. In Proceedings of the ACL-08: HLT, Columbus, OH, USA, 15–20 June 2008; pp. 1039–1047.
6. Lu, J.; Ng, V. Conundrums in event coreference resolution: Making sense of the state of the art. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 1368–1380.
7. Lu, J.; Ng, V. Event Coreference Resolution: A Survey of Two Decades of Research. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; International Joint Conferences on Artificial Intelligence Organization: Stockholm, Sweden, 2018; pp. 5479–5486. [CrossRef]
8. Alsaedi, N.; Burnap, P.; Rana, O. Automatic summarization of real world events using twitter. In Proceedings of the International AAAI Conference on Web and Social Media, Cologne, Germany, 17–20 May 2016; Volume 10, pp. 511–514.
9. Kompan, M.; Bieliková, M. Content-based news recommendation. In *LNBIP*; Lecture Notes in Business Information Processing; Springer: Berlin/Heidelberg, Germany, 2010; Volume 61, pp. 61–72. [CrossRef]
10. Pradhan, S.S.; Ramshaw, L.; Weischedel, R.; MacBride, J.; Micciulla, L. Unrestricted coreference: Identifying entities and events in ontonotes. In Proceedings of the International Conference on Semantic Computing (ICSC), Irvine, CA, USA, 17–19 September 2007; pp. 446–453. [CrossRef]
11. Taylor, A.; Marcus, M.; Santorini, B. The Penn treebank: An overview. In *Treebanks*; Springer: Dordrecht, The Netherlands, 2003; pp. 5–22.
12. Kingsbury, P.R.; Palmer, M. From treebank to propbank. In Proceedings of the Language Resources and Evaluation LREC, Las Palmas, Spain, 29–31 May 2002; pp. 1989–1993.
13. Kingsbury, P.; Palmer, M. Propbank: The next level of treebank. In Proceedings of the Treebanks and lexical Theories, Växjö, Sweden, 14–15 November 2003; Volume 3.
14. *ACE English Annotation Guidelines for Events (v5.4.3)*; Linguistics Data Consortium: Philadelphia, PA, USA, 2008.
15. Mitamura, T.; Yamakawa, Y.; Holm, S.; Song, Z.; Bies, A.; Kulick, S.; Strassel, S. Event Nugget Annotation: Processes and Issues. In Proceedings of the 3rd Workshop on EVENTS: Definition, Detection, Coreference and Representation, Denver, CO, USA, 4 June 2015; Association for Computational Linguistics: Denver, CO, USA, 2015; pp. 66–76. [CrossRef]
16. Cybulska, A.; Vossen, P. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, 26–31 May 2014; pp. 4545–4552.
17. Bejan, C.; Harabagiu, S. Unsupervised Event Coreference Resolution with Rich Linguistic Features. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, 11–16 July 2010; pp. 1412–1422. [CrossRef]

18. Song, Z.; Bies, A.; Strassel, S.; Riese, T.; Mott, J.; Ellis, J.; Wright, J.; Kulick, S.; Ryant, N.; Ma, X. From Light to Rich ERE: Annotation of Entities, Relations, and Events. In Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT, Denver, CO, USA, 4 June 2015; ACL: Denver, CO, USA, 2015; pp. 89–98.
19. Eirew, A.; Cattan, A.; Dagan, I. WEC: Deriving a large-scale cross-document event coreference dataset from Wikipedia. *arXiv* **2021**, arXiv:2104.05022.
20. Minard, A.L.; Speranza, M.; Urizar, R.; van Erp, M.; Schoen, A.; van Son, C. MEANTIME, the NewsReader Multilingual Event and Time Corpus. In Proceedings of the 10th Language Resources and Evaluation Conference (LREC), Portorož, Slovenia, 23–28 May 2016; European Language Resources Association (ELRA): Portorož, Slovenia, 2016; p. 6.
21. Rahman, A.; Ng, V. Supervised models for coreference resolution. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–7 August 2009; pp. 968–977.
22. Chen, C.; Ng, V. SinoCoreferencer: An End-to-End Chinese Event Coreference Resolver. *Lrec* **2014**, *2*, 4532–4538.
23. Cybulska, A.; Vossen, P. Translating Granularity of Event Slots into Features for Event Coreference Resolution. In Proceedings of the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, Denver, CO, USA, 4 June 2015; Association for Computational Linguistics: Denver, CO, USA, 2015; pp. 1–10. [CrossRef]
24. Nguyen, T.H.; Meyers, A.; Grishman, R. New York University 2016 System for KBP Event Nugget: A Deep Learning Approach. In Proceedings of the TAC, Gaithersburg, MD, USA, 14–15 November 2016; p. 7.
25. De Langhe, L.; De Clercq, O.; Hoste, V. Investigating Cross-Document Event Coreference for Dutch. In Proceedings of the Fifth Workshop on Computational Models of Reference, Anaphora and Coreference, Gyeongju, Republic of Korea, 16–17 October 2022.
26. Joshi, M.; Levy, O.; Weld, D.S.; Zettlemoyer, L. BERT for coreference resolution: Baselines and analysis. *arXiv* **2019**, arXiv:1908.09091.
27. Kantor, B.; Globerson, A. Coreference resolution with entity equalization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 673–677.
28. Joshi, M.; Chen, D.; Liu, Y.; Weld, D.S.; Zettlemoyer, L.; Levy, O. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 64–77. [CrossRef]
29. Lu, J.; Ng, V. Learning Antecedent Structures for Event Coreference Resolution. In Proceedings of the Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference, Cancun, Mexico, 18–21 December 2017; pp. 113–118.
30. Raghunathan, K.; Lee, H.; Rangarajan, S.; Chambers, N.; Surdeanu, M.; Jurafsky, D.; Manning, C. A Multi-Pass Sieve for Coreference Resolution. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, USA, 9–11 October 2010; Association for Computational Linguistics: Cambridge, MA, USA, 2010; pp. 492–501.
31. Lu, J.; Ng, V. Event Coreference Resolution with Multi-Pass Sieves **2016**. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; p. 8.
32. Liu, Z.; Araki, J.; Hovy, E.H.; Mitamura, T. Supervised Within-Document Event Coreference using Information Propagation. In Proceedings of the LREC, Reykjavik, Iceland, 26–31 May 2014; pp. 4539–4544.
33. Choubey, P.K.; Huang, R. *Event Coreference Resolution by Iteratively Unfolding Inter-Dependencies Among Events*; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 2124–2133. [CrossRef]
34. Lu, J.; Venugopal, D.; Gogate, V.; Ng, V. Joint inference for event coreference resolution. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 3264–3275.
35. Chen, C.; Ng, V. Joint Inference over a Lightly Supervised Information Extraction Pipeline: Towards Event Coreference Resolution for Resource-Scarce Languages. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AR, USA, 12–17 February 2016; pp. 2913–2920.
36. Araki, J.; Mitamura, T. Joint Event Trigger Identification and Event Coreference Resolution with Structured Perceptron. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; Association for Computational Linguistics: Lisbon, Portugal, 2015; pp. 2074–2080. [CrossRef]
37. Colruyt, C.; De Clercq, O.; Desot, T.; Hoste, V. EventDNA: A dataset for Dutch news event extraction as a basis for news diversification. *Lang. Resour. Eval.* **2022**, 1–33. [CrossRef]
38. Vermeulen, J. newsDNA: Promoting News Diversity: An Interdisciplinary Investigation into Algorithmic Design, Personalization and the Public Interest (2018–2022). In Proceedings of the ECREA 2018 Pre-Conference on Information Diversity and Media Pluralism in the Age of Algorithms, Lugano, Switzerland, 31 October 2018.
39. De Langhe, L.; De Clercq, O.; Hoste, V. Constructing a cross-document event coreference corpus for Dutch. *Lang. Resour. Eval.* **2022**, 1–30. [CrossRef]
40. Cybulska, A.; Vossen, P. *Guidelines for ECB+ Annotation of Events and Their Coreference*; Technical Report NWR-2014-1; VU University Amsterdam: Amsterdam, The Netherlands, 2014.
41. Desot, T.; De Clercq, O.; Hoste, V. 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), Varna, Bulgaria, 1–3 September 2021; pp. 346–357.
42. Van Noord, G.J. At Last Parsing Is Now Operational. In Proceedings of the Actes de la 13ème Conférence sur le Traitement Automatique des Langues Naturelles, Leuven, Belgium, 30 June 2006.
43. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.

44. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
45. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
46. Acheampong, F.A.; Nunoo-Mensah, H.; Chen, W. Transformer models for text-based emotion detection: A review of BERT-based approaches. *Artif. Intell. Rev.* **2021**, *54*, 5789–5829. [CrossRef]
47. Lin, C.; Miller, T.; Dligach, D.; Bethard, S.; Savova, G. A BERT-based universal model for both within-and cross-sentence clinical temporal relation extraction. In Proceedings of the 2nd Clinical Natural Language Processing Workshop, Minneapolis, MI, USA, 7 June 2019; pp. 65–71.
48. Chan, Y.H.; Fan, Y.C. A recurrent BERT-based model for question generation. In Proceedings of the 2nd Workshop on Machine Reading for Question Answering, Hong Kong, China, 4 November 2019; pp. 154–162.
49. De Vries, W.; van Cranenburgh, A.; Bisazza, A.; Caselli, T.; van Noord, G.; Nissim, M. Bertje: A dutch bert model. *arXiv* **2019**, arXiv:1912.09582.
50. Oostdijk, N.; Reynaert, M.; Hoste, V.; Schuurman, I. *The Construction of a 500-Million-Word Reference Corpus of Contemporary Written Dutch*; Springer Publishing Company Incorporated: Berlin/Heidelberg, Germany, 2013; pp. 219–247. [CrossRef]
51. Ordelman, R.; de Jong, F.; van Hessen, A.; Hondorp, G. TwNC: A Multifaceted Dutch News Corpus. *ELRA Newsl.* **2007**, *12*, 4–7.
52. Allan, J. *Topic Detection and Tracking*; Kluwer Academic Publishers: Norwell, MA, USA, 2002; pp. 1–16.
53. Allan, J. *Topic Detection and Tracking: Event-based Information Organization*; Springer Publishing Company Incorporated: Berlin/Heidelberg, Germany, 2012.
54. Boykin, S.; Merlino, A. Machine learning of event segmentation for news on demand. *Commun. ACM* **2000**, *43*, 35–41. [CrossRef]
55. Chen, T.; He, T. Xgboost: Extreme Gradient Boosting. R Package Version 0.4-2. 2015, Volume 1. Available online: https://cran.microsoft.com/snapshot/2017-12-11/web/packages/xgboost/vignettes/xgboost.pdf (accessed on 13 January 2023).
56. Chang, K.W.; Samdani, R.; Rozovskaya, A.; Sammons, M.; Roth, D. Illinois-Coref: The UI system in the CoNLL-2012 shared task. In Proceedings of the Joint Conference on EMNLP and CoNLL-Shared Task, Jeju Island, Republic of Korea, 12–14 July 2012; pp. 113–117.
57. Vilain, M.; Burger, J.D.; Aberdeen, J.; Connolly, D.; Hirschman, L. A model-theoretic coreference scoring scheme. In Proceedings of the Sixth Message Understanding Conference (MUC-6), Columbia, MA, USA, 6–8 November 1995.
58. Moosavi, N.S.; Strube, M. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 632–642.
59. Pradhan, S.; Luo, X.; Recasens, M.; Hovy, E.; Ng, V.; Strube, M. Scoring coreference partitions of predicted mentions: A reference implementation. In Proceedings of the Conference Association for Computational Linguistics, Baltimore, MA, USA, 22–27 June 2014, Volume 2014, p. 30.
60. Luo, X. On coreference resolution performance metrics. In Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, BC, Canada, 6–8 October 2005; pp. 25–32.
61. Stoyanov, V.; Gilbert, N.; Cardie, C.; Riloff, E. Conundrums in Noun Phrase Coreference Resolution : Making Sense of the State-of-the-Art. *Differences* **2009**, 656–664. [CrossRef]
62. Poot, C.; van Cranenburgh, A. A benchmark of rule-based and neural coreference resolution in Dutch novels and news. *arXiv* **2020**, arXiv:2011.01615.
63. Colruyt, C. Event Extraction: What Is It and What's Going on (14/03 Draft). 2018. Available online: https://www.netowl.com/what-is-event-extraction (accessed on 13 January 2023).
64. Atefeh, F.; Khreich, W. A survey of techniques for event detection in twitter. *Comput. Intell.* **2015**, *31*, 132–164. [CrossRef]
65. Wang, L.; Cardie, C.; Marchetti, G. Socially-informed timeline generation for complex events. *arXiv* **2016**, arXiv:1606.05699.
66. Gottschalk, S.; Demidova, E. EventKG–the hub of event knowledge on the web–and biographical timeline generation. *Semant. Web* **2019**, *10*, 1039–1070. [CrossRef]
67. Saggion, H. Automatic summarization: An overview. *Rev. Fr. Aise Linguist. Appl.* **2008**, *13*, 63–81. [CrossRef]
68. De Langhe, L.; De Clercq, O.; Hoste, V. Towards Fine (r)-grained Identification of Event Coreference Resolution Types. *Comput. Linguist. Neth. J.* **2022**, *12*, 183–205.
69. Clark, K.; Khandelwal, U.; Levy, O.; Manning, C.D. What does bert look at? an analysis of bert's attention. *arXiv* **2019**, arXiv:1906.04341.
70. Mikolov, T.; Grave, E.; Bojanowski, P.; Puhrsch, C.; Joulin, A. Advances in pre-training distributed word representations. *arXiv* **2017**, arXiv:1712.09405.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.