Leveraging syntactic parsing to improve event annotation matching

Camiel Colruyt, Orphée De Clercq, Véronique Hoste
LT^3, Language and Translation Technology Team
Ghent University
Groot-Brittanniëlaan 45, 9000 Ghent, Belgium
firstname.lastname@ugent.be

Abstract

Detecting event mentions is the first step in event extraction from text and annotating them is a notoriously difficult task. Evaluating annotator consistency is crucial when building datasets for mention detection. When event mentions are allowed to cover many tokens, annotators may disagree on their span, which means that overlapping annotations may then refer to the same event or to different events. This paper explores different fuzzy matching functions which aim to resolve this ambiguity. The functions extract the sets of syntactic heads present in the annotations, use the Dice coefficient to measure the similarity between sets and return a judgment based on a given threshold. The functions are tested against the judgments of a human evaluator and a comparison is made between sets of tokens and sets of syntactic heads. The best-performing function is a head-based function that is found to agree with the human evaluator in 89% of cases.

1 Introduction

The extraction of event descriptions from text has been the subject of many research efforts in the last decades (Vossen, 2016; Peng et al., 2015; Aguilar et al., 2014). Downstream tasks such as event coreference have also been studied (Lu and Ng, 2018, 2017; Araki and Mitamura, 2015). More specifically, *event mention extraction* refers to the task of identifying spans in the text that mention certain real-world events, as well as extracting given features of that mention. For example, the sentence “A car bomb exploded in central Baghdad”, according to the ACE guidelines (noa, 2008), contains a mention of an event of the type *Conflict.Attack*.

The difficulties of annotating events have been extensively discussed. Conceptually, events are difficult to define, as they are open to interpretation and may be worded in idiosyncratic ways (Vossen et al., 2018). Poor recall – i.e., human annotators not consistently recognising that events occur – is an acknowledged issue in event mention studies (Mitamura et al., 2015; Inel and Aroyo, 2019). Many datasets work with a fixed set of labels, representing the different semantic categories an event mention can belong to, but the choice of labels can be ambiguous. The reliability of such datasets has therefore been questioned (Vossen et al., 2018). Despite this ambiguity, annotation projects usually assume that there is a ground truth to event extraction, and trust a small number of annotators to discover it. Crowdsourcing event annotation can relax the search for a ground truth and reflect the ambiguity of the task more closely, as well as provide an implicit consistency-checking mechanism. (Inel and Aroyo, 2019), for instance, use crowdsourcing to validate and extend the annotations of select event and time datasets.

Another issue is that, as different research projects design conceptualizations geared to specific tasks, the resulting data sets are specialized to a certain degree: they over- or underrepresent certain genres event types. Models trained on this data can perform well within that range but transfer poorly to work with uncurated data in a real-world context (Araki and Mitamura, 2018). This relationship between task, dataset and performance has been examined in e.g. Grishman (2010).

In many event annotation schemas (RED (O’Gorman et al., 2016), ACE (Peng et al., 2016), ECB+ (Cybulska and Vossen, 2014), MEANTIME (Minard et al., 2016)), event mention detection relies on the identification of a single-token lexical trigger for the event. In “A car bomb exploded in central Baghdad”, the token *exploded* is annotated as the trigger (noa, 2008). The 2014 Knowledge Base Population (KBP) track of the Text Analysis Conference (TAC) introduced Event

Proceedings of the First Workshop on Aggregating and Analysing Crowdsourced Annotations for NLP, pages 15–23
Hong Kong, China, November 3, 2019. ©2019 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

Nuggets, an event description that allows tagging multiple tokens as the event trigger. Multi-token triggers allow annotators to more easily navigate cases in which multiple words could be chosen as a trigger, e.g. “hold a meeting”, “serve a sentence”, and they can be continuous or discontinuous (Mitamura et al., 2015).

In this paper event triggers are further generalized to comprise entire clauses, encompassing all arguments to the event, in order to alleviate the same type of issues we found when annotating Dutch news text (see Section 3 for a description of this project). However, using expanded triggers introduces the possibility that annotators may identify the same *events* while disagreeing on the exact *span* of the event mentions, either through error or a different interpretation of the mention’s scope. It is therefore important to monitor the quality and consistency of annotations. This is all the more true given the expensive and time-consuming nature of annotation. Inter-annotator agreement studies must be conducted carefully to gauge how human annotators interpret and apply certain guidelines.

In a consistency study, F-score is measured by determining in how far different annotators recognize the same event mentions in a text. When annotations from both annotators cover the same span, this is trivial. However, when annotators annotate the same event but mark different spans, a fuzzy matching mechanism is necessary to recognize that both annotations refer to the same event. In this paper, we explore such fuzzy matching methods and compare two different methods: matching the similarity between token sets as done in previous work (Liu et al., 2015), versus relying on syntactic head sets which are obtained through parsing. We find that using heads leads to results that lie closer to the judgment of a human evaluator.

The remainder of this paper is organized as follows: in Section 2 previous research on evaluating event mention annotations is described. Section 3 introduces the event dataset and annotation procedure. Section 4 explains the methodology for matching annotation pairs and in Section 5, the results are described. Section 6 concludes this paper and offers prospects for future work.

2 Related work

Event recognition was introducted as a task in the ACE program in 2004. Event mention recall was not evaluated directly; rather, scoring happened on the level of *events* rather than *event mentions* (such that one event is a bucket of multiple mentions referring to the same event). A mapping between gold mentions and system output mentions was a prerequisite for scoring (Doddington et al., 2004). The same is true for the Event Mention Detection (VMD) task in ACE 2005 (National Institute of Standards and Technology, 2005). The ACE 2005 corpus thereafter became widely used in event detection studies. Because triggers consist of single-word tokens, testing recall is straightforward. Li et al. (2013) and Li et al. (2014), for instance, treat matches as correct if their offsets (span) and event subtype match exactly. F1 is used to score performance overall.

Event nugget detection, and with it event triggers that consist of more than one word, were introduced as a task in the TAC KBP track in 2014. Liu et al. (2015) proposes a method to evaluate nugget recall which enables fuzzy matching for annotations with non-perfectly-overlapping spans. In this work the Dice coefficient is used to measure the set similarity between the tokens covered by each annotation, which turns out to be the same as F1 score. System mentions are mapped to gold standard mentions by selecting the gold-system pair with the highest Dice coefficient score. An overall matching score is produced by considering other features of the event annotation. Mitamura et al. (2015) uses this method to assess the consistency of annotation in the 2014 TAC KBP corpus. This paper uses the same idea and takes the Dice coefficient to map annotations from different annotators, but applies it to the sets of heads of mentions. Additionally, Dice-based methods are applied to token sets to examine the advantages of using heads over tokens. However, the triggers in event nuggets still mostly consist of single tokens (Mitamura et al., 2015) and multi-token triggers are kept minimally short. In our task, event mentions span several tokens by default, making a straightforward comparison difficult.

3 Dataset and annotations

In this paper we report on the inter-annotator agreement (IAA) study of an event annotation task carried out in the framework of the #NewsDNA project, a large interdisciplinary research project on news diversity. To allow for automatic Dutch event extraction, training data is required and the objective is to annotate over 1,500 news articles coming from major Flemish publishers. In a first phase, 34 articles were annotated by four linguists to allow for the IAA study. For this paper, 4 additional articles were annotated and used to evaluate annotation matching methods.

The articles were annotated with information on events, entities and IPTC media topics[1]. Our event annotations are structurally similar to ACE/ERE events. The spans are marked and augmented with information on arguments, type and subtype (following a typology close to that of ACE/ERE), realis properties (polarity, tense and modality) and prominence (whether the event is a main event of an article or a background event). As mentioned before, the focus of this paper lies on the annotation of event mention spans, which can comprise entire verbal clauses or nominal constructions. Figure 1 shows an example of a fully-annotated event mention from the IAA set carried out in the WebAnno annotation tool (Yimam et al., 2014).

While annotation guidelines were devised describing the constraints of annotation as closely as possible, there is a large gray area in which annotators may interpret event span boundaries differently. For instance, in Table 1, examples 1 and 2 show cases where annotators disagreed on including descriptive clauses in the event mention. Matching annotations may also diverge due to annotation errors; in example 3, a punctuation mark was annotated by mistake. Consequently, matching annotations with different spans occur frequently in the IAA set. When we say two annotations *match*, we mean they intend to mark the same *mention* of the event in the text. (This differs from coreference, which aims to match different *mentions* that refer to the same event.) Contrasting with this, there are cases of overlapping annotations which do *not* refer to the same event, as in Table 2.

In order to conduct an IAA study, it is necessary to match the annotation of different annotators correctly. Such a matching function must mimic human judgment in finding that the span pairs in Table 1 match, but the pair in Table 2 does not. In this paper, we explore possibilities for matching these annotations based on set similarity functions of the syntactic heads of the spans.

4 Matching annotations

In this section, we describe the methods we used to evaluate different annotation matching functions. We call a *matching function* a function that takes two annotations as input. It returns True if they match and False if they do not.

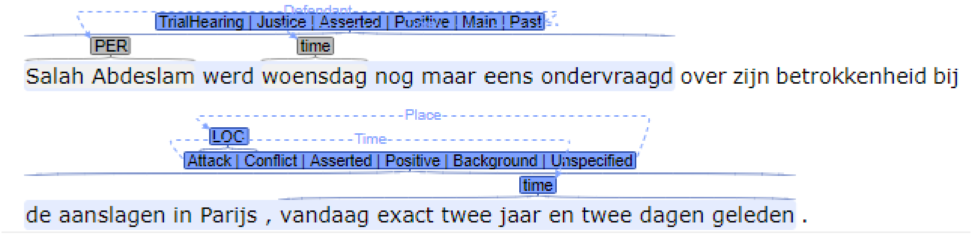
4.1 Extracting the syntactic heads of annotations

We described a need for matching functions that emulate human judgment. Intuitively, we consider annotations to match if the "semantic core" of their constructions agree. Given a pair of annotations like [*There were several violations — There were several violations severe enough to talk about a breach of confidence*], we would roughly identify *"violations"* as the core element of the event described. If two annotations share the same semantic core, we consider them to match. Conversely, in the pair [*There were several violations severe enough to talk about a breach of confidence — a breach of confidence*], the semantic cores are different and the annotations do not match. The *"breach of confidence"* is not the focus of the first event mention.

We intuitively correlated the idea of semantic cores with the syntactic heads of the mention. In order to derive these syntactic heads, the state-of-the-art Alpino dependency parser for Dutch was used (van Noord, 2006). Using these parse trees, we extracted the set of *head tokens* from each annotation. We define a head token as any token that has "HD" as a dependency label in its node, or whose node is a child (directly or not) of a HD node. In an Alpino parse, the label HD is used to mark the verb in any verbal construction, the nominal core of a nominal construction, the preposition in a prepositional phrase and the core elements of adverbial groups (van Noord et al., 2018). Table 2 shows the syntactic heads extracted in this way from two non-matching annotations. Figure 2 shows a visualization of the syntactic tree obtained from the same example.[2]

[1] https://iptc.org/standards/media-topics/

[2] Visualized via `http://nederbooms.ccl.kuleuven.be/eng/alpinotree`.


Defendant
TrialHearing | Justice | Asserted | Positive | Main | Past
PER
time
Place
LOC
Time
Attack | Conflict | Asserted | Positive | Background | Unspecified
time

Salah Abdeslam werd woensdag nog maar eens ondervraagd over zijn betrokkenheid bij

de aanslagen in Parijs , vandaag exact twee jaar en twee dagen geleden .

Figure 1: Example of two fully annotated event mentions in the IAA corpus.
Translation: *Salah Abdeslam was questioned once again on Wednesday about his complicity in the attacks in Paris, which took place exactly two years and two days ago.*

Annotator A	Annotator B
(1) Trump geeft VN kritiek bij eerste speech *Trump criticizes UN during first speech*	Trump geeft VN kritiek *Trump criticizes UN*
(2) Er waren herhaaldelijke inbreuken *There were several violations*	Er waren herhaaldelijke inbreuken die zwaar genoeg zijn om te spreken van een vertrouwensbreuk *There were several violations severe enough to talk about a breach of confidence*
(3) De Amerikaanse president Donald Trump heeft voor het eerst de Algemene Vergadering van de Verenigde Naties toegesproken. *The American president Donald Trump has addressed the General Assembly of the United Nations for the first time.*	De Amerikaanse president Donald Trump heeft voor het eerst de Algemene Vergadering van de Verenigde Naties toegesproken *The American president Donald Trump has addressed the General Assembly of the United Nations for the first time*

Table 1: Examples of matching overlapping mentions.

4.2 Match function candidates

The matching functions we explore in this paper score an annotation pair by the set similarity of their syntactic head sets. At the same time, a series of functions which use the set similarity of the token sets is also evaluated in order to test the relative advantage of using head over token sets, if any.

The similarity score is defined as the Dice coefficient between the two sets. This is the same measure used by Liu et al. (2015) to measure the similarity between event annotation token sets, and is equivalent to F1. The Dice coefficient returns a score between 0 and 1, where 1 equals complete overlap.

$$
\begin{aligned}
Dice(S_i, S_j) &= \frac{2|S_i S_j|}{|S_i| + |S_j|} \\
&= \frac{2}{|S_i|/|S_i S_j| + |S_j|/|S_i S_j|} \\
= F1(S_i, S_j) &= \frac{2}{1/P + 1/R}
\end{aligned}
$$

We set various thresholds over this score to achieve boolean functions. For instance, given an annotation pair (a_i, b_i) with a head set Dice coefficient of 0.6, a Dice-based matching function with a threshold of 0.5 will return True, and one with a threshold of 0.8 will return False. Finding a Dice-based function that emulates human behaviour means finding the right threshold at which the Dice function will maximally agree with the human evaluator. Note that at a threshold of 0, a Dice function will always return True, and at threshold 1 it will return False for anything less than exact overlap between the sets.

We designed and evaluated two baseline functions and two families of Dice-based functions, for a total of 40 matching functions:

- A random function that returns True or False with equal likelihood, which can be considered a baseline.
- A function which returns True if the tokens of both annotations match after punctuation has been removed.

Annotator A	Annotator B
De Amerikaanse president Donald Trump heeft voor het eerst de Algemene Vergadering van de Verenigde Naties toegesproken *The American president Donald Trump addressed the General Assembly of the United Nations for the first time*	de Algemene Vergadering van de Verenigde Naties *the General Assembly of the United Nations*
[president, heeft, vergadering, van, verenigde naties, toegesproken]	[vergadering, van, verenigde naties]

Table 2: Overlapping but non-matching mentions and the sets of their syntactic heads.

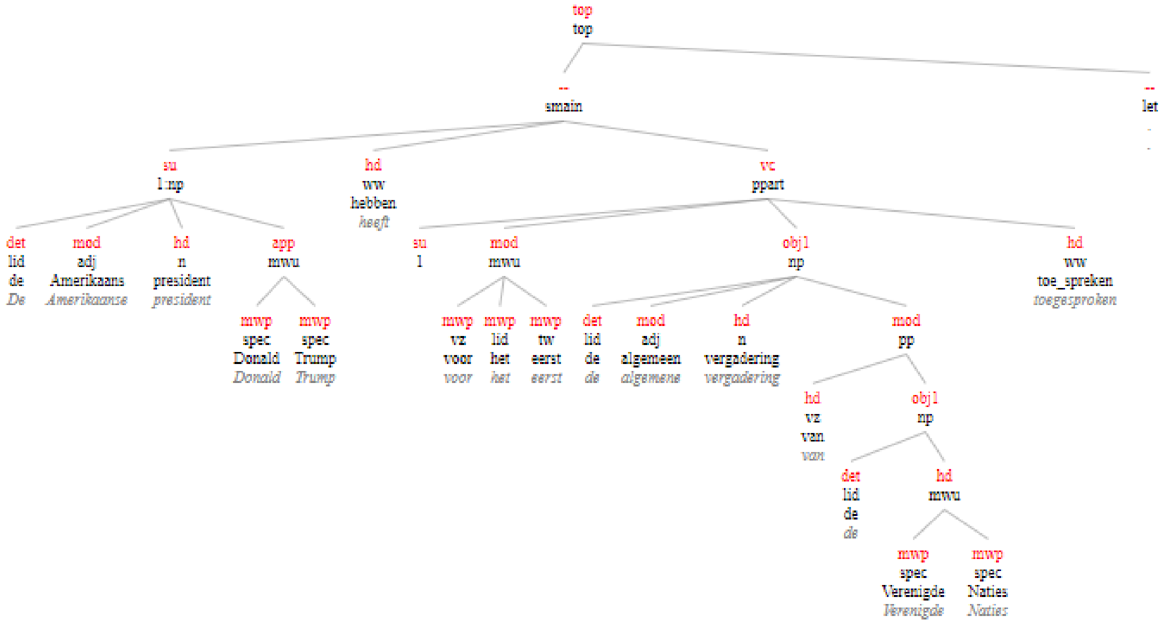


Figure 2: Visualization of an Alpino dependency tree used to extract syntactic heads.

- A series of 19 Dice-based threshold functions which run over the head sets of the annotation pair. Thresholds were chosen from the range $[0, 1]$ with a step of 0.05 (so the series of $[0.05, 0.1, 0.15, ..., 0.95, 1]$).
- The same series of 19 Dice functions operating over token sets.

4.3 The process of evaluating matching functions

To test the performance of a matching function, we compare its output to the judgment of a human evaluator. To this purpose, four additional news articles were selected and annotated by the same four annotators from the IAA set and judged by an independent human evaluator.[3] The goal of this evaluation is to discover how close the candidate fuzzy matching functions we designed come to matching the judgment of a human evaluator, and which of these candidates should be applied in the IAA study proper. In this section, we describe the actual evaluation process itself which consists of four phases.

(i) Collecting annotations Given two annotators, A and B, we attempt to match the annotations they made over a given sentence s. That is, for every pair of partial annotations over s, a human evaluator judges whether the pair matches or not; the fuzzy matching functions are run over the same pair and scored on how they agree with the human evaluator. Let a_i be A's annotations over s and b_i B's annotations over s. All pairs of annotations over a_i and b_i are collected: $[a_1b_1, a_1b_2, a_1b_3, ...]$. The Dice coefficient is symmetrical,

[3] As we will explain later, there is a negative skew in this dataset which must be taken into account.

such that the results over $(a_1 b_1)$ are equal to the results over $(b_1 a_1)$. Accordingly, only one of each such pairs is included.

(ii) Overlap filter A first filter function checks the overlap between the two annotated strings in each pair. If they overlap perfectly, the pair is counted as a matching pair without going through human evaluation or fuzzy matching. If they do not overlap at all, they are counted as not matching. At the end of the first filter, the annotation pairs are sorted in three sets: O_T, the set of perfectly overlapping annotations; O_P, the set of partially overlapping annotations, and O_F, the set of non-overlapping annotations.

(iii) Human and system evaluations Human and system evaluation is only performed on O_P. The human evaluator and each fuzzy matching function judge each pair as matching or not matching. We denote the set of pairs that the human evaluator judges as matching to H_T when they match and to H_F when they do not. For a given candidate matching function C, we obtain similar sets C_T and C_F.

(iv) Scoring The judgment of the human evaluator is taken as the gold standard answer. Each function is then scored based on its agreement with the human evaluation. For a candidate function C, we count the number of pairs on which it agrees with the human annotator as n_{Agr}. The score of a function is the ratio of the number of agreements over the total number of partially overlapping annotations.

$$n_{Agr} = |H_T \cap C_T| + |H_F \cap C_F|$$
$$S = \frac{n_{Agr}}{|O_P|}$$

It reads as a number between 0 and 1, where 1 represents total agreement with the human evaluator.

5 Results

In total, 182 annotation pairs were collected. Table 3 summarizes set statistics after phases (i) to (iii). Of the 182 pairs, 44 overlapped perfectly and 77 not at all. Of the remaining 61 partially overlapping pairs, the human annotator counted 44 pairings as false and 16 as true. The negative skew indicates that most overlapping annotations are not the same events annotated differently, but simply mentions of different events that happen to overlap (e.g. Table 2).

Set of pairs	Count
Total	182
O_T	44
O_P	61
O_F	77
H_T	17
H_F	44

Table 3: Annotation pair statistics over the four evaluation documents.

Matching function	Score
Random match (baseline)	0.43
Punctuation removed (baseline)	0.79
Dice 0.0 (head or token)	0.28
Dice 0.75, 0.8 (head)	**0.89**
Dice 1.0 (head)	0.79
Dice 0.75-0.90 (token)	0.85
Dice 1.0 (token)	0.72

Table 4: Results of the different matching functions

Table 4 reports on the different matching functions (Section 4.2). The reported random score is the average over three runs, and obtains a score of 0.43. This can be read as agreement with the human evaluator in 43% of cases. We single out a few results from the Dice functions. Dice 0.0 on heads or tokens always returns True and agrees in 28% of cases. Dice 1.0 on tokens always returns False by definition, since it expects perfect overlap of token sets but is only run on partially overlapping sets. It scores 0.72. The negative skew in the dataset is evident in these two results. Dice 1.0 on heads returns True only if the heads sets of the annotation pairs match exactly. It intuitively works well in cases where there is little diversity in annotated spans, and disagreements revolve around insignificant elements or punctuation marks. It scores 0.79. As a comparison, we tested a baseline which return True if the tokens of each annotation match exactly after punctuation has been discarded, which also scores 0.79.

Table 5 gives the scores for Dice functions on each threshold. The small size of the set prevents us from tuning the Dice threshold on a separate development set; we therefore present scores for all functions. The strictly best-performing function was found to be Dice on heads with thresholds 0.75 or 0.8, with a top score of 0.89, indicated in bold in Table 4. The Dice functions on tokens plateau between thresholds 0.75 to 0.90 with a score of 0.85. We therefore found head-based matching to outperform token-based matching by 4 percentage points.

Since this scoring method omits the judgments made in the overlap filter phase (phase (ii) in Section 4.3), we also calculated a *full score* which reflects the function's performance if the judgments of phase (ii) are taken into account as well

$$S_{Full} = \frac{n_{Agr} + |O_T| + |O_F|}{|O_P| + |O_T| + |O_F|}$$

With this measure, Dice 0.75-0.80 on heads scores 0.96, and Dice 0.75-0.90 on tokens scores 0.95. The relative advantage of using heads is less apparent in these measures, since partial matches constitute only 61 of 182 total annotation pairs. In other words, about two thirds of pairs are judged during phase (ii), such that the effect of fuzzy matching is less pronounced.

Dice threshold	On heads	On tokens
1.0	0.79	0.72
0.95	0.79	0.80
0.9	0.82	**0.85**
0.85	0.85	**0.85**
0.8	**0.89**	**0.85**
0.75	**0.89**	**0.85**
0.7	0.70	0.84
0.65	0.69	0.84
0.6	0.67	0.85
0.55	0.67	0.74
0.5	0.66	0.54
0.45	0.66	0.51
0.4	0.62	0.44
0.35	0.56	0.44
0.3	0.54	0.39
0.25	0.44	0.36
0.2	0.39	0.31
0.15	0.34	0.31
0.1	0.34	0.28
0.05	0.34	0.28
0.0	0.28	0.28

Table 5: Scores of all Dice functions.

6 Discussion and conclusion

This pilot study evaluated a set of 40 fuzzy matching functions to match event annotations over different annotators for use in consistency studies. In our corpus, annotations span several tokens by default, and there is a considerable gray zone where annotators may mark the same event but disagree on the exact span. To perform an inter-annotator study in this context, it is necessary to have fuzzy matching functions that can determine whether overlapping annotations match or refer to different events. An evaluation set of 182 potentially matching annotation pairs was devised and functions were tested against the judgment of a human evaluator.

Our intuition was that matching functions which leverage the syntactic heads of the annotation, presumed to be the "semantic centers" of the span, would outperform token-based fuzzy matching methods. Head-based functions were indeed found to perform best overall, with a score of 0.89 for the Dice 0.75-0.80 functions on heads. Therefore, we show that the fuzzy matching functions we devised emulate human judgment more closely than the baseline, and that functions using syntactic heads perform better than token-based functions. Given these results we will proceed with the best-performing head-based matching function for our corpus IAA study.

As future work we wish to conduct a more fine-grained exploration of the resulting dependency trees, which could further benefit matching by restricting the concept of syntactic heads. We took as heads the nodes tagged as "HD" in the dependency tree or the descendants of these nodes; these include the prepositions in prepositional phrases and the cores of adverbial groups. Filtering out these heads would further reduce the head sets of annotations to essential elements. Additionally, heads could be limited to nodes appearing in the top few levels of the tree. Conversely, relying more on parsing information implies a risk of error propagation, where errors made in parsing percolate into matching. We also wish to investigate which, if any, impact this has on the results.

Acknowledgements

We thank the reviewers for their helpful comments. This work was supported by #NewsDNA (Ghent University, Belgium), an interdisciplinary research project aimed at developing and testing an algorithm that uses news diversity as a key driver for personalised news recommendation. http://newsdna.ugent.be.

References

2008. *ACE English Annotation Guidelines for Events (v5.4.3)*. Linguistics Data Consortium.

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.

Jun Araki and Teruko Mitamura. 2015. Joint Event Trigger Identification and Event Coreference Resolution with Structured Perceptron. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2074–2080, Lisbon, Portugal. Association for Computational Linguistics.

Jun Araki and Teruko Mitamura. 2018. Open-Domain Event Detection using Distant Supervision. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 878–891, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The Automatic Content Extraction (ACE) Program Tasks, Data, and Evaluation. In *Proceedings of LREC*.

Ralph Grishman. 2010. The Impact of Task and Corpus on Event Extraction Systems. *Lrec*, pages 2928–2931.

Oana Inel and Lora Aroyo. 2019. Validation methodology for expert-annotated datasets: Event annotation case study. In *2nd Conference on Language, Data and Knowledge, LDK 2019*, page 12. Schloss Dagstuhl- Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing.

Qi Li, Heng Ji, Yu Hong, and Sujian Li. 2014. Constructing information networks using one single model. *Methods on Natural Language Processing (EMNLP2014)*, pages 1846–1851.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint Event Extraction via Structured Prediction with Global Features. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82.

Zhengzhong Liu, Teruko Mitamura, and Eduard Hovy. 2015. Evaluation Algorithms for Event Nugget Detection : A Pilot Study. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 53–57, Denver, Colorado. Association for Computational Linguistics.

Jing Lu and Vincent Ng. 2017. Joint Learning for Event Coreference Resolution. pages 90–101. Association for Computational Linguistics.

Jing Lu and Vincent Ng. 2018. Event Coreference Resolution: A Survey of Two Decades of Research. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 5479–5486, Stockholm, Sweden. International Joint Conferences on Artificial Intelligence Organization.

Anne-Lyse Minard, Manuela Speranza, Ruben Urizar, Marieke van Erp, Anneleen Schoen, and Chantal van Son. 2016. MEANTIME, the NewsReader Multilingual Event and Time Corpus. In *Proceedings of the 10th language resources and evaluation conference (LREC 2016)*, page 6, Portoro, Slovenia. European Language Resources Association (ELRA).

Teruko Mitamura, Yukari Yamakawa, Susan Holm, Zhiyi Song, Ann Bies, Seth Kulick, and Stephanie Strassel. 2015. Event Nugget Annotation: Processes and Issues. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 66–76, Denver, Colorado. Association for Computational Linguistics.

National Institute of Standards and Technology. 2005. The ACE 2005 (ACE05) Evaluation Plan. Technical report.

G. J. van Noord. 2006. At Last Parsing Is Now Operational.

Gertjan van Noord, Ineke Schuurman, and Gosse Bouma. 2018. Lassy Syntactische Annotatie (Revision 19456). page 208.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer Event Description: Integrating event coreference with temporal, causal and bridging annotation. *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 47–56.

Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015. A Joint Framework for Coreference Resolution and Mention Head Detection. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 12–21, Beijing, China. Association for Computational Linguistics.

Haoruo Peng, Yangqi Song, and Dan Roth. 2016. Event Detection and Co-reference with Minimal Supervision. *Emnlp*, pages 392–402.

Piek Vossen. 2016. *Newsreader public summary*, volume 31.

Piek Vossen, Filip Ilievski, Marten Postma, and Segers Roxane. 2018. Don't Annotate, but Validate: a Data-to-Text Method for Capturing Event Data. In *LREC2018, Myazaki*.

Seid Muhie Yimam, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. 2014. Automatic Annotation Suggestions and Custom Annotation Layers in WebAnno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96, Baltimore, Maryland. Association for Computational Linguistics.