# Tagging Terms in Text

## A Supervised Sequential Labelling Approach to Automatic Term Extraction

*Ayla Rigouts Terryn, Véronique Hoste, Els Lefever*
*LT³ Language and Translation Technology Team, Ghent University*

**Abstract**
As with many tasks in natural language processing, automatic term extraction (ATE) is increasingly approached as a machine learning problem. So far, most machine learning approaches to ATE broadly follow the traditional hybrid methodology, by first extracting a list of unique candidate terms, and classifying these candidates based on the predicted probability that they are valid terms. However, with the rise of neural networks and word embeddings, the next development in ATE might be towards sequential approaches, i.e., classifying each occurrence of each token within its original context. To test the validity of such approaches for ATE, two sequential methodologies were developed, evaluated, and compared: one feature-based conditional random fields classifier and one embedding-based recurrent neural network. An additional comparison was added with a machine learning interpretation of the traditional approach. All systems were trained and evaluated on identical data in multiple languages and domains to identify their respective strengths and weaknesses. The sequential methodologies were proven to be valid approaches to ATE, and the neural network even outperformed the more traditional approach. Interestingly, a combination of multiple approaches can outperform all of them separately, showing new ways to push the state-of-the-art in ATE.

**Keywords**
Terminology, automatic term extraction, sequential labelling

## 1 Introduction

Automatic term extraction (ATE; sometimes called automatic term recognition or ATR) is the task of identifying specialised vocabulary in collections of domain-specific texts. The results can either be used directly to facilitate term management for, e.g., terminologists and translators, or as a preprocessing step for other tasks within natural language processing (NLP), ranging from automatic indexing (Koutropoulou and Efstratios 2019) to aspect-based sentiment analysis (De Clercq et al. 2015). In the former case, ATE is usually considered a semi-automatic process that requires human validation, since it is such a difficult task that cannot yet be perfectly automated. One of the main difficulties for ATE lies in the ambiguous distinction between terms and general language. This is difficult even for humans, so capturing the nature of terms in a set of clear rules for the automation of the task is extremely challenging.

The traditional, hybrid approach to ATE, which still reaches state-of-the-art results, typically uses linguistic information to extract an initial list of candidate terms (CTs) from a specialised corpus, and filters and ranks this list based on statistical metrics. The result will be a list of unique CTs with the most likely ones ranked at the top. As with most research in NLP, it has become common practice to apply machine learning to the problem of ATE. No single feature performs well for ATE in all contexts (performance is often highly dependent on domain,

corpus size, language, etc.), so there is a proven benefit to combining multiple features (see, e.g., Dobrov & Loukachevitch, 2011). Therefore, while rule-based approaches are far from obsolete, the ability of machine learning to effectively combine many features poses a considerable advantage. There are many variations in methodologies, but most machine learning approaches to ATE have broadly followed the traditional approach, i.e., training a classifier to predict whether a given unique CT is a valid term or not. However, ATE can also be interpreted as a sequential labelling task, where each token in a running text is classified as (part of) a term or not. With this strategy, no lists of CTs are extracted, and instead, each occurrence of each token is classified within its original context. This strategy has been employed for related tasks such as Named Entity Recognition (Goyal, Gupta, and Kumar 2018) and automatic keyword recognition (Alami Merrouni, Frikh, and Ouhbi 2020), but only rarely for ATE.

The goal was to investigate sequential labelling approaches to ATE. To do so, two alternative sequential methodologies have been developed: a feature-based approach using a conditional random fields (CRF) classifier, and a neural approach using only embeddings. Both are extensively compared and evaluated on the ACTER dataset (Annotated Corpora for Term Extraction Research) (Rigouts Terryn, Hoste, and Lefever 2020). Scores are calculated based on the sequential results, and the sequential labels are also used to extract a list of unique CTs and calculate f1-scores against the non-sequential gold standard (GS). After presenting related research in section 2, section 3 will offer a summary of the dataset and of the conversion of the original annotations to a suitable dataset for the classifiers. Section 4 is dedicated to the system descriptions. In section 5, the experimental setup is explained and the results of both sequential systems with different configurations are summarised. These results are discussed in more detail in section 6. A final error analysis with examples is presented in section 7, before concluding with a summary of the results and ideas for future research.

# 2 Related Research

## 2.1 Machine Learning Approaches

As mentioned, there are many non-machine learning approaches to ATE that still obtain state-of-the-art results, such as TermoStat (Drouin 2003), TExSIS (Macken, Lefever, and Hoste 2013), Termolator (Meyers et al. 2018), and TermSuite (Cram and Daille 2016). The initial linguistic and statistical approaches tend to be combined into a hybrid methodology. A typical pipeline would start with the linguistic preprocessing of the corpus, i.e., tokenisation, lemmatisation, part-of-speech (POS) tagging, etc. This may also include syntactic chunking or parsing. CTs can then be extracted from the text based on predefined POS patterns (sometimes also using syntactic information). Many systems focus on nominal terms (Kageura and Marshman 2019), filter out stopwords, apply a frequency threshold, and/or restrict the minimum and maximum CT length. This initial list of CTs can then be filtered and sorted with statistical termhood and unithood measures (Kageura and Umino 1996). Termhood indicates how relevant a term is to the domain, whereas unithood signifies the cohesion between the different tokens of a multi-word term. Usually, one statistical measure is chosen to sort the result and only the *n* (%) highest ranked CTs are kept, or only those above a certain threshold value. Simple voting strategies can be used to combine multiple measures (Vivaldi and Rodríguez 2001), though this is not common. The rise of machine learning in NLP offered a new way to efficiently combine multiple metrics for ATE. The first experiments with

(supervised) machine learning for ATE were often based on the traditional method and still start with a rule-based approach to extract CTs based on POS patterns or syntactic information, or by selecting all n-grams with a maximum length and minimum frequency. The machine learning aspect only comes into play during the second step, when features are calculated for the extracted CTs and an algorithm can learn the optimal combination of features from annotated training data, to classify these CTs as either terms or not terms. Often, the extracted CTs can be ranked based on the classifier's predicted probability that they are valid terms, so the results are presented in the same format as the traditional approach, i.e., a list of CTs with the most probable true terms at the top. Recently, there have been attempts to step away from this CT-based approach, in favour of a sequential labelling approach. Rather than classifying unique CTs that have already been extracted from the text, such an approach will label tokens in the text itself. Each token is analysed in its context and classified as a potential (part of) a term or not. Accordingly, each occurrence of each token is treated separately, as opposed to the traditional approach in which all occurrences of a CT were grouped and treated as a single instance. Since this project concerns machine learning methodologies, the related research will focus on those studies specifically. Since studies on sequential approaches to ATE are still very rare, non-sequential machine learning approaches (classification of CTs) will be addressed as well.

## 2.2 Evaluation

The accepted evaluation metrics for ATE are precision, recall, and f1-score, which compare the extracted CTs to a predefined list of GS terms. They are defined as follows:

$$\text{precision} = \frac{correctly\ extracted\ terms\ (true\ positives)}{all\ extracted\ CTs\ (true\ positives + false\ positives)}$$

$$\text{recall} = \frac{correctly\ extracted\ terms\ (true\ positives)}{all\ terms\ in\ the\ corpus\ (true\ positives + false\ negatives)}$$

$$\text{f1-score} = 2 * \frac{precision * recall}{precision + recall}$$

Due to the difficulty and cost of creating completely annotated corpora, recall cannot always be calculated and researchers resort to alternatives, such as relative recall (Amjadian et al. 2018), average precision (Fedorenko, Astrakhantsev, and Turdakov 2013), and precision@rank (Zhang, Petrak, and Maynard 2018). While there are many datasets – which have been annotated to various degrees and with varying accuracy – few have been used for multiple studies, except GENIA (Kim et al. 2003) in the domain of biomedicine, and ACL RD-TEC (Qasemizadeh and Handschuh 2014; Qasemizadeh and Schumann 2016) in the field of computational linguistics. Nevertheless, even when the same scores are calculated on these same datasets, the GS and the way the scores are calculated may still differ. For instance, there might be restrictions on term length (e.g., between 1 and 5 tokens (Yuan, Gao, and Zhang 2017)) or term frequency (e.g., only CTs that occur 10+ times (Hätty, Dorna, and Schulte im Walde 2017)), or the calculation of the scores might count partial matches as correct (Bay et al. 2020). Therefore, it is nearly impossible to get a fair idea of state-of-the-art scores. In answer to this issue, a recent shared task on ATE (Rigouts Terryn et al. 2020) allowed participants to develop and fine-tune a system based on provided training data, and all submissions were evaluated on the same test data. Despite the limited timeframe and number of participating teams (five), the results illustrate the difficulty of the task well, with modest f1-scores ranging

between 13.2% and 46.7%. Even with the help of the latest machine techniques, there remains a lot of room for improvement in the field of ATE.

### 2.3 Features

The use of machine learning allowed researchers to broaden the types of information that could be used as clues to find terms. POS patterns and termhood and unithood measures are still important but are now often supplemented with other types of features. As discussed in previous work (Rigouts Terryn, Hoste, and Lefever 2021), examples range from simple features about the shape of the CT, like length and capitalisation, to features that rely on external resources, and more complex features based on the use of language models and topic models. An especially noteworthy evolution is the use of embeddings. For instance, the best scoring system in the TermEval shared task (Hazem et al. 2020) contrasts a feature-based approach to a deep neural network using BERT models (Devlin et al. 2019), eventually concluding that the latter performs better in English, but that results for both approaches are comparable in French.

When embeddings are used for ATE, these are often pre-trained embeddings, potentially fine-tuned during classification. Pre-trained GloVe embeddings[1] have been used in several studies (Amjadian et al. 2018; Kucza et al. 2018; Zhang, Petrak, and Maynard 2018) and word2vec (Mikolov, Yih, and Zweig 2013) has been used to train domain-specific embeddings, usually with the CBOW and/or skip-gram architectures (Amjadian et al. 2018; Bay et al. 2020; Wang, Liu, and McDonald 2016). Some studies have attempted to combine general embeddings and domain-specific embeddings. The first one (Amjadian et al. 2016; 2016) does so for the formerly mentioned English corpus on mathematics with 1.1M+ tokens, another (Hätty, Schlechtweg, and Dorna 2020) on German corpora, which, even after preprocessing and removal of all non-content words, still contain at least 0.7M words per domain. Another example is a Canadian-English corpus of 1.5M+ tokens on the topic of unwanted behaviours from potential employees (Drouin, Morel, and Homme 2020). These examples immediately illustrate a first issue with this methodology: they require huge corpora. One of the smallest corpora used to train domain-specific embeddings for ATE still counts 368k words (Bay et al. 2020), in which case it was used in combination with a statistical measure and required a seed set of validated terms, so that new CTs would only be retained if they were close to one of the validated terms. Another aspect these studies have in common is that they only extract single-word terms (unigrams); while it is possible to train n-gram-based embeddings, this can become even more computationally expensive.

Despite the required computational power and the need for very large corpora, the combination of general and domain-specific embeddings remains a potentially promising strategy for ATE. In the English study (Amjadian et al. 2018), the general and domain-specific vectors are concatenated and used as input for a Multi-Layer Perceptron (MLP). The German study (Hätty, Schlechtweg, and Dorna 2020) goes a step further and tests two (neural) approaches to combine both embeddings and map them into the same space. Both approaches to combine the two vectors were found to work better than a simple concatenation of general and domain-specific

[1] https://nlp.stanford.edu/projects/glove/

vectors, and any strategy using both vectors performs better than using either the general or the domain-specific one by itself.

### 2.4 Sequential Approaches

In sequential labelling tasks, each token is classified within its original context. This is often done with an IOB labelling scheme, (Habibi et al. 2017), where each first token of a relevant entity is tagged as B (Beginning), each subsequent token within that entity as I (Inside), and tokens that are not part of any relevant entity are tagged as O (Outside). Sometimes such labelling is also performed at character-level (e.g., Kucza et al., 2018). IOB labelling schemes do not always allow encoding of nested annotations. For instance, suppose the sequence *a supervised machine learning approach* contains two terms which need to be encoded: *supervised machine learning* and *machine learning*; then the IOB labels could be *a[O] supervised[B] machine[B] learning[I] approach[O]*, but that could be interpreted as the annotations of *supervised* and *machine learning*, not of *supervised machine learning*. Therefore, sometimes the B label is only used for the beginning of nested entities (not the beginning of annotations following an O label). In that case, the sequence could be tagged as *a[O] supervised[I] machine[B] learning[I] approach[O]*, which would allow the correct extraction off both terms. Nevertheless, this does not always suffice, so there are more complex annotation schemes as well. Instead of an IOB scheme, some use BILOU (Kucza et al. 2018) or IOBES (Rokas, Rackevičienė, and Utka 2020), which can, e.g., have separate tags for tokens that form a single-word term by themselves and the last token of a multi-word term. Despite the added labels, this still does not allow the detection of all complex nested annotations. Consider, for instance, the example in Figure 1. Even the nested annotation of *heart* in *heart failure* is problematic with IOB labels. Therefore, a common approach is to only annotate the longest possible sequence (Kucza et al. 2018). In the example, this would mean only annotating *heart failure with preserved ejection fraction* and *HFpEF* with sequential labels. The nested annotations (*heart, heart failure, preserved ejection fraction, ejection fraction*) might then only be found if they occur separately elsewhere in the corpus, not nested in other annotations.

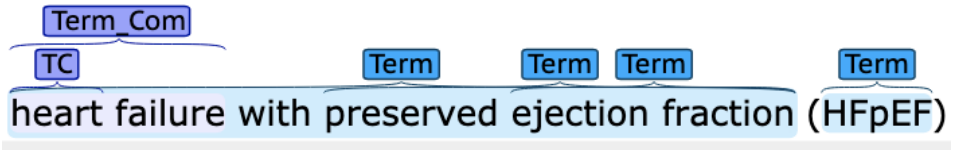


*Figure 1: Example of complex recursive (nested) annotation in BRAT interface*

In related tasks, such as biomedical NER, sequential labelling approaches are relatively common. For instance, Habibi et al. (2017) show how a generic deep learning method with word2vec word embeddings (Mikolov, Yih, and Zweig 2013) is often able to outperform state-of-the-art methods. They use an LSTM-CRF architecture for this purpose (Long Short-Term Memory network - Conditional Random Field). For ATE, sequential methodologies are still very rare. The first (to the best of their, and our knowledge) to employ a sequential labelling approach in the context of ATE are Kucza at al. (2018). The reported scores are macro-averaged precision, recall, and f1-scores for all five labels (no list of CTs is extracted). They compare recurrent neural networks (RNNs): LSTMs versus GRUs (gated recurrent units),

using both pre-trained word embeddings and character embeddings (both end-to-end trained and pre-trained), and train and test their models on the GENIA and ACL RD-TEC corpora. Some important findings were that preprocessing data (lowercasing and removing punctuation) leads to slightly worse performance, and that scores were drastically reduced for out-of-domain testing (training on GENIA and testing on ACL RD-TEC or the reverse) compared to training and testing within the same corpus. The top macro-averaged f1-score for in-domain testing was 86.89%, versus only 48% for out-of-domain testing, in which case character embeddings outperform word embeddings. Another attempt at sequential labelling for ATE was performed for the Irish language, using the IOB labelling scheme and reporting scores per label (McCrae and Doyle 2019), and one for Lithuanian cybersecurity terms with FastText and BERT embeddings (Rokas, Rackevičienė, and Utka 2020). Earlier linguistic approaches to ATE, like one by Bourigault (1993) implemented in the LEXTER tool (Bourigault 1992) could also be considered as a type of sequential approach to ATE, as it uses noun phrases to find boundaries of terms in texts. However, since the current research focuses on machine learning approaches, this work will not be discussed in more detail. In conclusion, sequential labelling seems to be a viable option for ATE, but it has not really been compared to the traditional approach yet and requires a lot more research. By comparison, word embeddings have been more extensively researched in this context but can also benefit from more comparative research with feature-based methods.

The current project attempts to contribute by contrasting and evaluating two types of sequential approaches: one feature-based, one neural with embeddings. Additionally, these are compared to the traditional approach, using a machine learning architecture with similar features as the sequential feature-based approach. All experiments are performed with the same dataset, which covers multiple domains and languages for a more robust evaluation. An in-depth error analysis is performed to identify the strengths and weaknesses of the approaches. This research demonstrates how sequential machine learning methodologies are valid approaches to ATE, which might be able to push the state-of-the-art.

## 3 Data

The ACTER 1.4 dataset (Rigouts Terryn, Hoste, and Lefever 2020) contains three annotated comparable corpora and one parallel corpus in three languages (English, French, and Dutch), and four domains (corruption (*corp*), equitation - dressage (*equi*), heart failure (*htfl*), and wind energy (*wind*)). The original annotations were made with the BRAT rapid annotation tool (Stenetorp et al. 2011) (see also screenshot in Figure 1) with four labels: Specific Terms, Common Terms, Out-of-Domain Terms, and Named Entities, which were defined based on their domain-specificity (how relevant is the term to the domain) and lexicon-specificity (how much expertise is required to know the term). More information on these categories can be found in the original paper or the annotation guidelines[2]. For the current project, a binary definition is needed (term vs. not term), so unless specifically mentioned otherwise,

[2] http://hdl.handle.net/1854/LU-8503113

annotations of all four labels are considered terms, i.e., positive instances. In total, the dataset counts 681,463 tokens (50,845 to 64,990 per corpus).

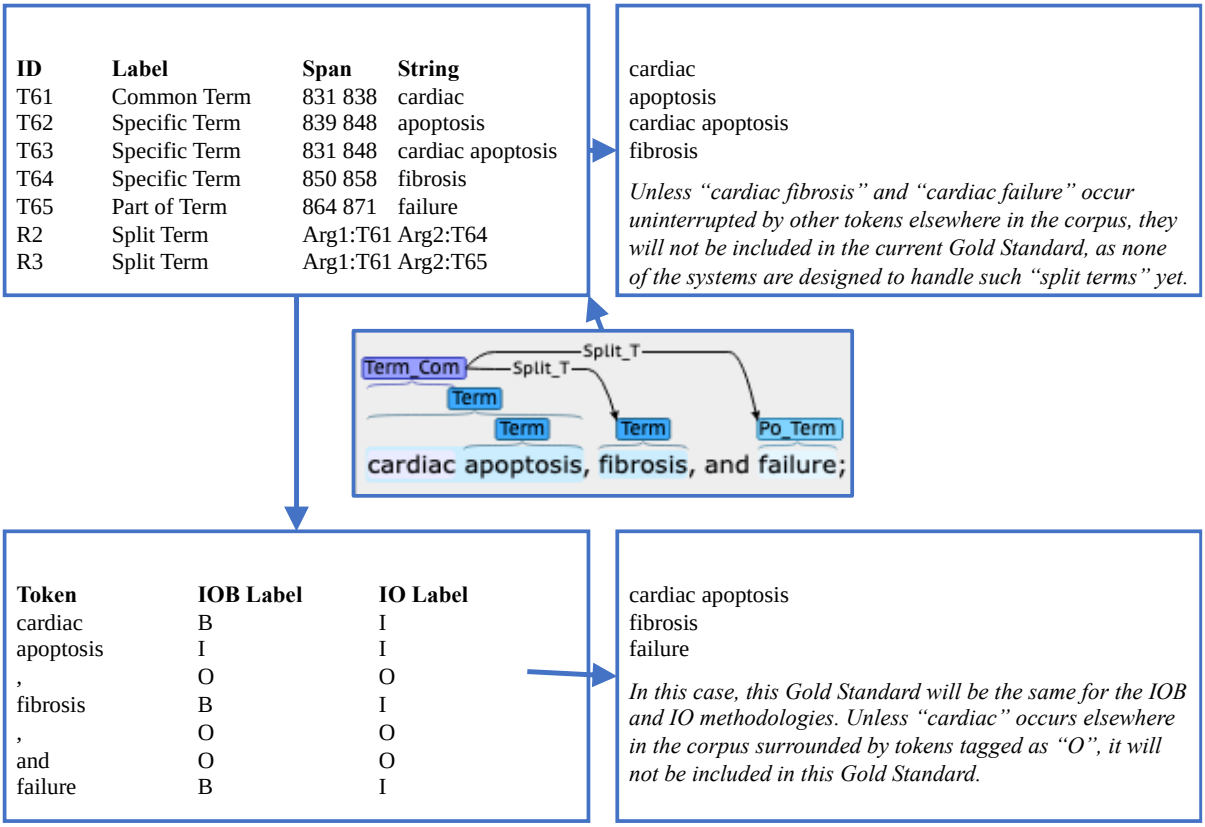


*Figure 2: Schematic of how the data, annotated in BRAT (middle screenshot), was converted into the different Gold Standards (GS)*

The BRAT annotations result in a separate file per text, with one annotation per line, identified by the indices of the first and last character of that annotation (top left box in Figure 2). To create a sequentially labelled dataset at token-level, rather than at character-level, the original annotations were mapped to tokens. Aligning the character-level annotations to tokens was relatively straightforward in most cases, especially since discontinuous terms (split terms, see annotation guidelines) were excluded from the dataset. However, there were two scenarios that were not quite as simple. The first concerns annotations that begin or end within a token, so that only a part of that token is annotated. While the guidelines specify that, generally, annotations should not be made within tokens, this is allowed for complex terms separated by a dash (e.g., *angiotensin-receptor blocker*, where *angiotensin* is annotated as a term). In cases where this led to only part of a token being included in any annotation, it was decided to consider the entire token as a valid part of a term, so it would get an I or B labelling. However, such cases are rare, since such annotations were often nested within longer annotations anyway, as in the example cited above.

The second difficulty concerns tokens that are part of discontinuous annotations, for instance in the case of ellipses (see Figure 2). The discontinuous terms cannot be included in the GS

(unless they occurred somewhere else without interruptions), since none of the methodologies in the current project are equipped to deal with such split terms yet. However, for a sequential methodology, it was deemed most logical to still tag these parts of terms as positive instances, since they are (at least partially) terminological. Deciding to tag tokens that are only partially annotated, and tokens that are part of discontinuous annotations as positive instances in the GS was most logical in a sequential setup, but not in the traditional methodology for ATE. In such a traditional approach, (candidate) terms are presented as a list of unique instances, so it would not be desirable to include such partially correct annotations. Therefore, the GS for the traditional approach (top right box in Figure 2), and the GS data used for the sequential approach (bottom left box in Figure 2), are not only presented in a different format, but also contain slightly different GS annotations.

Capturing all complex nested annotations with a simple sequential annotation scheme is impossible. Therefore, the commonly used IOB scheme was applied to encode only the longest possible sequences of annotations, without considering any nested annotations. The first token of a sequence was tagged as B and all subsequent tokens within the annotation were tagged as I (even if they were the beginning of a nested annotation). While IOB labels are common practice, it was hypothesised that a simple binary scheme, where B and I labels are combined (IO instead of IOB), could also be interesting. The bottom left box in Figure 2 shows an example with both tagging schemes. When two terms are not separated by a non-terminological token, the IO scheme cannot represent these annotations as accurately as the IOB scheme. Nevertheless, a binary classification task is usually easier to model, which might compensate for the potential loss in accuracy.

Since one of the goals of this project is to contrast traditional (non-sequential) to sequential ATE, the data also had to be compared somehow. Therefore, sequentially labelled data was converted to lists of unique instances, so that both the GS and the results of sequential approaches could be compared more thoroughly to those of the traditional, non-sequential approach. An example is the transformation of the sequential GS data in the bottom left box in Figure 2 to a list of unique GS terms in the bottom right box of Figure 2. As discussed, there will be three differences when comparing the traditional GS (top right box in Figure 2) to a GS in the same format extracted from the sequential data (bottom right box in Figure 2): (1) annotations that only ever occur nested within other annotations will not be included in the GS based on sequential data, (2) annotations of tokens that are only partially annotated will not be included in the traditional GS, and (3) tokens that are only tagged as part of a discontinuous (split) term will not be included in the traditional GS either.

To discover how well the IOB and IO labelled data align with the traditional, non-sequential GS, and how much accuracy is lost between the IOB and IO approaches, the IOB and IO GS based on the sequentially labelled data were evaluated compared to the original list of standard terms, used for more traditional (non-sequential) ATE. Lower scores in this comparison are not necessarily bad, since both approaches have a different purpose, i.e., presenting a list of unique terms versus indicating all terms in a running text. Therefore, differences are expected and do not mean one approach or dataset is superior. Nevertheless, it is important to be aware of the differences and such a comparison is a practical way to find them. Extracting all terms from the IOB labelled GS results in an average f1-score of 93.4% compared to the traditional GS; for the IO GS, the f1-score drops to 85.1%. This is a considerable difference, so the results of further experiments will have to determine whether this drop in potential accuracy with the IO approach can be compensated by the comparatively easier setup for the classification task

(binary vs. multiclass classification). For the experiments in the remainder of this contribution, whenever the output of sequential systems is evaluated against the traditional GS, these scores should be considered an approximation of the upper bounds.

A final note on the dataset is that, beyond tokenisation, no changes were made to the original data, and no normalisation was performed. The characteristics of terms are so diverse that almost any filter risks changing or removing a term. For instance, function words and special characters were not filtered out since they occur in terms like *quality of life* and *β-blocker*. Capitalisation was maintained in the sequential dataset and/or used as a feature, but whenever terms are presented as lists of unique instances, all data is lowercased, so that instances which are identical apart from their capitalisation are combined.

# 4 System Description

## 4.1 CRFSuite Feature-based Sequential ATE

The feature-based sequential system uses CRFSuite (Okazaki 2007), an implementation of CRFs for Python, to apply a linear-chain CRF with Adaptive Regularisation Of Weight Vector (AROW) (Crammer, Kulesza, and Dredze 2009). CRFSuite's standard settings were used and the variance and gamma hyperparameters were optimised through grid search with 10-fold cross-validation on the training set. The optimisation was based on macro-averaged f1-scores for all labels. Since the goal was not only to compare this feature-based method to another embedding-based sequential method, but also to a traditional (non-sequential) machine learning approach to ATE, the features were largely based on the ones used in the latter (HAMLET) (Rigouts Terryn, Hoste, and Lefever 2021). The main difference is the inclusion of contextual and string-based features. All texts are linguistically preprocessed with LeTs Preprocess (tokenisation, lemmatisation, POS-tagging, chunking, and NER) (van de Kauter et al. 2013), and all language-dependent POS tags are mapped to a shared set of 26 tags (Rigouts Terryn, Hoste, and Lefever 2021) based on Universal Dependencies (Petrov, Das, and McDonald 2012). These 26 tags (*standard* POS) were also mapped to a more coarse-grained set of 8 tags (*simple* POS). The reference corpora, used for frequency-based and statistical features, are Wikipedia dumps in all languages and newspaper reference corpora, all limited to 10M tokens. The news corpora were News on Web for English (Davies 2017), the Gigaword corpus for French (Graff, Mendonça, and DiPersio 2011), and news-related subcorpora of openSONAR for Dutch (Oostdijk et al. 2013). Statistical features that require a reference corpus are all calculated twice, compared to both types of reference corpora.

Some of the features only pertain to the token itself, whereas other features look at all occurrences of that token in the corpus. In most cases, only tokens with the same full form and capitalisation are combined (regardless of POS), but a few features consider five other variations of the token: (1) same lowercased full form, (2) same normalised form, (3) same full form and simple POS, (4) same lowercased full form and POS, (5) same lowercased lemma and POS. All statistical features are all explained in the work of Astrakhantsev at al. (2015), except for Vintar's termhood measure, which can be found in her own work (Vintar 2010). As can be seen in Table 1, there are 100 features in total, split into seven categories.

| Category | Feature description | # fts. |
|---|---|---|
| **Token** | token itself | 1 |
| **Context** | previous and next 3 tokens | 6 |
| | simple POS tag of previous and next 3 tokens | 6 |
| | standard POS tag of previous and next 3 tokens | 6 |
| | NER tag of previous and next 3 tokens | 6 |
| | chunking information from previous and next 3 tokens | 6 |
| | CT occurs before, after, between, or nowhere near brackets | 4 |
| **Linguistic** | simple POS | 1 |
| | standard POS | 1 |
| | NER information | 2 |
| | chunking information | 2 |
| | # possible POS tags (simple & standard) for all occurrences of token | 2 |
| | probability of current POS tag (simple & standard) for token | 2 |
| | token is in stopword list[3] | 1 |
| **Shape** | alphanumeric characteristics of token | 1 |
| | capitalisation of token | 1 |
| | # capitalisation options for all occurrences of token and probabilities | 5 |
| | # characters, digits, and special characters in token | 3 |
| | prefix and suffix (first and last 3 characters) of token | 2 |
| | suffix of lemmatised form of token | 1 |
| **Frequency** | frequency and document frequency in domain-specific corpus | 2 |
| | frequency and document frequency in reference corpora | 4 |
| **Statistical** | domain pertinence vs. reference corpora | 2 |
| | domain relevance vs. reference corpora | 2 |
| | domain specificity vs. reference corpora | 2 |
| | log-likelihood ratio vs. reference corpora | 2 |
| | relevance vs. reference corpora | 2 |
| | TF-IDF | 1 |
| | Vintar's termhood measure vs. reference corpora | 2 |
| | Weirdness vs. reference corpora | 2 |
| **Variation** | 5 variants of token | 5 |
| | domain specificity vs. Wikipedia corpus for all variants | 5 |
| | Vintar's termhood measure vs. news corpus for all variants | 5 |
| | frequency in specialised corpus for all variants | 5 |

*Table 1: Features for CRFSuite system for sequential ATE*

[3] The ISO stopwords were used for all languages: https://github.com/stopwords-iso

### 4.2 FlairNLP Neural, Embedding-based Sequential ATE

The neural approach was implemented using the FlairNLP framework (Akbik et al. 2019), an open source library for sequential NLP tasks in Python. It allows a straightforward implementation of a Recurrent Neural Network (RNN) for sequential labelling tasks in NLP, using the embeddings offered through PyTorch (Paszke et al. 2019). We use the standard biLSTM-CRF architecture, with a single hidden layer of size 512 and the AdamW optimiser (Kingma and Ba 2015) with weight decay regularisation from Loschilov & Hutter (2019). The ACTER corpora are very small (51k-65k tokens per corpus), and the goal, for now, is not to build the best possible model, but rather to compare different approaches and identify the strengths and weaknesses of these methodologies more generally. Therefore, PyTorch's pre-trained embeddings (on large, general corpora) were used, and no domain-specific embeddings are trained. To enable fair comparisons across languages, comparable embeddings had to be available for each of the three languages in the corpus. This excluded GloVe or ELMo embeddings (Peters et al. 2018) for instance. The three types of embeddings that were used all incorporate even subword or character-level information, which is thought to be helpful for tasks that include a lot of rare words (which is likely the case in our specialised, domain-specific corpora). None of the embeddings were specifically tuned for the task and only the standard settings were used.

Multiple embeddings were tested, starting with the **Flair embeddings**, since we worked within the FlairNLP framework. These "contextual string embeddings" (Akbik, Blythe, and Vollgraf 2018, 1638) are obtained with a neural, character-based language model and can incorporate both previous and next context by stacking the "backward" and "forward" embeddings in the FlairNLP framework, that is designed to easily combine (stack) embeddings. They achieved state-of-the-art results in sequence labelling for named entity recognition, which made them a promising first choice for ATE. The pre-trained embedding are trained on a 1 billion word newspaper corpus (for English embeddings), French Wikipedia (for French embeddings), the Dutch texts of the Wikipedia OPUS corpus (Wołk and Marasek 2014) (for Dutch embeddings), and the JW300 corpus, a "parallel corpus of over 300 languages with around 100 thousand parallel sentences per language pair on average" (Agić and Vulić 2019, 3204) for the multilingual embeddings.

**FastText** embeddings (Bojanowski et al. 2016) were chosen as the next logical option of popular and often successful embeddings that are available for all languages in the project and that also incorporate subword information. We used the embeddings that are pre-trained on Common Crawl (for all languages).

Finally, the hugely successful transformer-based architectures are supported in FlairNLP as well, through HuggingFace (Wolf et al. 2020), so **BERT embeddings** (Devlin et al. 2019) could be tested as well. For English, "bert-base-cased" was used, for French CamemBERT (Martin et al. 2020), and for Dutch BERTje (de Vries et al. 2019). Both Flair and BERT models were available as multilingual embeddings too, so these were included in the evaluation as well. BERT multilingual embeddings are trained on monolingual Wikipedia corpora in the top 104 languages on Wikipedia, without any markers to indicate the difference between the languages. They are shown to generalise well cross-lingually, especially between similar languages (Pires, Schlinger, and Garrette 2019).

### 4.3 HAMLET Machine Learning Approach to Traditional Hybrid ATE

The HAMLET system (Rigouts Terryn et al. 2019; Rigouts Terryn, Hoste, and Lefever 2021) is not the focus of this research, but it is used for comparison. It is a machine learning approach to ATE, also based on the ACTER corpus and with features like those of the CRFSuite feature-based approach, but according to the traditional approach to ATE. CTs are first extracted based on the POS patterns of the annotated training data, and features are calculated to classify each CT as either a term or not, with a confidence score, using a Random Forest Classifier (RFC) in Scikit Learn (Pedregosa et al. 2011). It reaches state-of-the-art results, usually with higher recall than precision. Whenever HAMLET is compared to the sequential approaches, they are trained an evaluated on the exact same corpora.

## 5 Experiments and Results

### 5.1 Experimental Setup

Despite the proven benefit of domain-specific training data, real-life applications will rarely have access to large, domain-specific, annotated datasets. Therefore, the strictest, but most realistic setting was chosen for the experiments: training on out-of-domain data and testing on a separate, unseen corpus in a different domain. For instance, when results are reported on the English heart failure corpus, the system has been trained on the three other English corpora (corruption, dressage, and wind energy). Per corpus, each experiment was repeated three times with identical data and settings, so results could be averaged over these three trials and provide an indication of standard deviation.

For sequential ATE, no consensus has been reached yet about the most appropriate metrics. A first option would be to use micro-averaged f1-scores of all labels, which, for a task like this, where each instance is given one label, would be the same as accuracy. Micro-averaging scores of multiple labels means that the average scores per label are multiplied by the number of instances that were assigned this label, before adding them and dividing them by the total number of instances in the dataset. In other words, micro-averaging scores over multiple labels considers how many instances each label covers. Conversely, macro-averaging scores of multiple labels would assign equal weights to all labels, regardless of how often each label is used. Using micro-averaged f1-scores of all labels for the current sequential ATE task would assign a disproportionate weight to the negative instances (O labels), which, on average, constitute around 81% of all tokens. Therefore, a classifier that predicts O for all tokens would reach a micro-averaged f1-score of 81%, despite not having detected a single term. Macro-averaging would be fairer because it would consider all labels equally, but for ATE, we are mostly interested in the scores of the positive labels. Consequently, it was decided to consider only the f1-scores of the positive labels (B and I). This strategy would also more closely resemble the reasoning behind the evaluation metrics for traditional ATE. In the case of the IO scheme, there is only a single positive label, so its f1-score did not need to be averaged. For the IOB data there are two positive labels that do not occur in equal proportions (13% B, and 6% I labels on average). Since these proportions are different, the micro-averaged f1-scores were calculated, i.e., considering the number of instances in each class before averaging.

Additionally, to allow comparison with traditional ATE and the traditional GS (see Figure 2), traditional precision, recall, and f1-scores were also calculated by extracting lists of unique CTs based on the assigned IO(B) labels and comparing those against the traditional GS. It should be emphasised that, as explained, this puts the sequential approaches at a minor disadvantage.

### 5.2 CRF Results

The obtained scores, averaged over all corpora, are represented in Table 2. Clearly, the two types of scores are very different: micro-averaged f1-scores for the positive labels in the sequential data are considerably better than f1-scores compared to the traditional, non-sequential GS. Conversely, f1-scores for the positive labels are much lower than, for instance, micro-averaged f1-scores for all labels (accuracy), which would be 83.0% and 85.1% for IOB and IO respectively (not shown in table).

| | **micro-averaged scores for positive label(s)** | | | | **scores compared to traditional GS** | | | |
|---|---|---|---|---|---|---|---|---|
| | **p** | **r** | **f1** | **σ of f1** | **p** | **r** | **f1** | **σ of f1** |
| **IOB** | 52.7 | 43.6 | 46.0 | 4.9 | 33.9 | 35.9 | 33.9 | 7.7 |
| **IO** | 66.4 | 53.9 | 57.0 | 5.7 | 33.8 | 36.5 | 33.6 | 7.3 |

*Table 2: Scores (as percentages) of IOB and IO CRFSuite systems, averaged over all corpora; standard deviation is calculated over three trials per corpus*

Considering the difficulty of the task, these scores are promising, but leave room for improvement. Scores for the IOB versus the IO system show the expected pattern: the binary (IO) approach reaches higher f1-scores on the sequential data than the IOB approach, but the f1-scores compared to the traditional GS are almost identical. This supports our hypothesis that, while the binary approach is a less accurate representation of terms in sequential data, this is at least partially compensated by the increased performance of the sequential classifier on the binary (IO) versus multi-label (IOB) task: in most (though not all) cases, sequential f1-scores are better for IO labelled data, but non-sequential, traditional f1-scores are similar for IO and IOB labelled data. This observation applies to experiments with the neural classifier as well. A final observation concerning these experiments with the CRF classifier is the large average standard deviation. The positive labels (I & B) represent less than 20% of all tokens, so relatively small differences in the results overall can lead to much larger disagreement in the scores. Average standard deviation for accuracy (of all labels) is considerably lower at 3.3% and 3.6%.

### 5.3 RNN results

As mentioned, the neural approach is tested with three types of embeddings: Flair, FastText, and BERT embeddings. For the latter two, both monolingual and multilingual embeddings are examined. The multilingual models are trained on all corpora, except the test corpus itself. This

means domain-specific data is included, but only in different languages. Furthermore, the Flair framework allows users to stack different embeddings, and the creators state that Flair embeddings might perform best when combined with others. Consequently, additional systems were trained with stacked Flair+BERT embeddings. The results are shown in Table 3, with the same metrics as for the CRFSuite model.

| | | | micro-averaged scores over positive label(s) | | | | scores compared to traditional GS | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | **p** | **r** | **f1** | **σ of f1** | **p** | **r** | **f1** | **σ of f1** |
| **monolingual embeddings** | **fast-Text** | **IOB** | 60.8 | 12.9 | 18.9 | 7.4 | 41.3 | 15.0 | 19.0 | 6.3 |
| | | **IO** | 69.2 | 23.4 | 32.0 | 10.1 | 37.1 | 21.6 | 24.4 | 6.5 |
| | **Flair** | **IOB** | 66.2 | 42.0 | 48.3 | 4.3 | 43.2 | 45.4 | 42.1 | 2.4 |
| | | **IO** | 74.8 | 52.3 | 58.4 | 3.0 | 41.6 | 47.6 | 42.3 | 1.4 |
| | **BERT** | **IOB** | 73.3 | 44.2 | 52.4 | 2.6 | 51.7 | 48.0 | **47.1** | 1.6 |
| | | **IO** | 80.5 | 50.4 | 58.8 | 2.9 | 49.4 | 47.8 | 45.8 | 1.6 |
| | **Flair + BERT** | **IO** | 78.7 | 50.0 | **59.7** | 3.9 | 43.7 | 38.4 | 39.6 | 1.2 |
| **multilingual embeddings** | **Flair** | **IOB** | 63.5 | 44.3 | 51.5 | 3.0 | 40.4 | 48.4 | 42.6 | 1.1 |
| | | **IO** | 71.8 | 52.5 | 58.1 | 2.8 | 38.1 | 47.4 | 40.8 | 1.3 |
| | **BERT** | **IOB** | 74.9 | 66.2 | 69.4 | 0.8 | 74.9 | 66.2 | 69.4 | 0.8 |
| | | **IO** | 81.3 | 71.4 | **75.2** | 0.9 | 81.3 | 71.4 | **75.2** | 0.9 |
| | **Flair + BERT** | **IO** | 81.0 | 71.1 | 74.9 | 0.9 | 81.0 | 71.1 | 74.9 | 0.9 |

*Table 3: Scores (as percentages) for RNN systems with different monolingual and multilingual embeddings, averaged over all corpora; standard deviation is calculated over three trials per corpus*

Table 3 reveals that results vary not only for different types of embeddings, but also depending on the evaluation metric. For instance, the monolingual stacked Flair+BERT embeddings reach the highest micro-averaged f1-score for positive labels, but the f1-score compared to the traditional GS is lower than all but the FastText models. Likewise, the relation between the IO and IOB labels is not straightforward: the IO models invariable get higher f1-scores on the sequential data, but this varies for the traditional f1-scores. Therefore, the intended purpose is a big factor in deciding which model is best suited for a project. For the sequential scores, precision is always higher than recall, especially for the FastText models, which reach very low recall. For all but the FastText models, standard deviation is considerably lower than for the feature-based models, especially for the (multilingual) BERT models. Results for the models that stack the best two types of embeddings (BERT and Flair) are not (much) better than the others, despite the increased computational cost.

Concerning the monolingual and multilingual models, not just the models are different between these experiments, but also the training data. Therefore, further experiments were performed to test the impact of that training data with BERT embeddings. Three additional experiments were performed with the multilingual BERT embeddings (only on IO labelled data). First, the model was trained on the same data as the monolingual systems (three out-of-domain corpora in the same language as the test corpus). Next, to test the impact of the amount of training data, it was trained on all nine out-of-domain corpora in all languages, excluding the two domain-specific corpora in the other languages. Finally, to test the impact of in-domain training data, it was trained on only the two in-domain corpora in the other languages. The results are reported in Table 4.

As expected, performance with multilingual BERT embeddings is similar to monolingual BERT embeddings when both are provided with the same training data (second row in Table 4). The final system (last row), trained only on in-domain data in other languages than the test corpus, performs marginally better than the system trained on all available data. While the difference is small and only applies to sequential scores, it is remarkable since the best-performing system only has access to two training corpora, which are a subset of the training data of the original multilingual system. This emphasises the importance of in-domain training data. By comparison, the amount of (out-of-domain) training data has much less impact. Comparing the second and third rows of results shows that adding six out-of-domain training corpora in other languages barely leads to any improvement.

In conclusion, BERT embeddings outperform both FastText and Flair embeddings for this task. The (multilingual) models benefit from in-domain training data, even when it is in other languages. However, since the goal of this project is to approximate a realistic setting where in-domain training data is unlikely to be available, further experiments will use only monolingual BERT embeddings, where the classifier is trained on three out-of-domain corpora in the same language as the test corpus. This also improves comparability with the results of the feature-based CRF model, which is not designed to use multilingual data.

| **Training data includes:** | | | **# training corpora** | **micro-averaged scores for positive label(s)** | | | | **scores compared to traditional GS** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **OOD data in same language** | **OOD data in other languages** | **In-dom. data in other languages** | | **p** | **r** | **f1** | **σ of f1** | **p** | **r** | **f1** | **σ of f1** |
| Yes | Yes | Yes | 11 | 81.3 | 71.4 | 75.2 | 0.9 | 47.7 | 61.3 | **52.7** | 0.7 |
| Yes | No | No | 3 | 79.5 | 50.4 | 59.1 | 3.2 | 46.5 | 47.6 | 45.1 | 1.6 |
| Yes | Yes | No | 9 | 80.8 | 50.8 | 59.7 | 2.6 | 49.1 | 47.8 | 45.9 | 1.4 |
| No | No | Yes | 2 | 79.1 | 74.8 | **76.6** | 1.3 | 41.8 | 60.9 | 49.3 | 1.0 |

*Table 4: Scores (as percentages) for model with multilingual Bert embeddings and binary IO labelling scheme with different training data*

# 6 Analyses and Discussion of Results

## 6.1 Choice of Experiments and Motivation

So far, the reported scores were averaged over three trials per corpus, and once again averaged over all corpora. Since it is not feasible to perform an in-depth analysis for all possible models and methodologies, a selection needed to be made. Instead of working with averages, only the best of three trials per corpus will be used, so that the actual output can be examined. Additionally, all analyses will continue with the RNNs, and the feature-based models will not be discussed in more detail for now. While both approaches reach comparable scores on the sequential data, the feature-based results are less stable (larger standard deviation), and scores compared to the traditional GS are better for the RNN. As discussed, the RNN will use monolingual BERT word embeddings, and will be trained on three out-of-domain corpora and evaluated on a held-out test corpus. To avoid double results for IOB and IO labelled data, only the latter was used for further experiments. For the current RNN with monolingual BERT embeddings specifically, the binary IO approach reaches slightly higher scores and the difference between the output of the IO and IOB systems is very small. On 97% of all tokens, the IO and IOB systems agree on either a positive (I or B) or negative label (O). In conclusion, all further experiments concern the best results out of three trials, per corpus, for an RNN with monolingual BERT embeddings, trained and evaluated on IO labelled data.

## 6.2 Results per Corpus

The results on each corpus for the RNN are reported in Table 5. As explained, these scores are slightly higher than in the previous tables because the previous tables reported on the averages over three trials, whereas the next tables all report the best scores out of three trials per experiment. The scores of HAMLET (Rigouts Terryn, Hoste, and Lefever 2021) on the same data are included in the same table and will be examined in more detail in section 6.3.

| | | **Sequential RNN with BERT embeddings** | | | | | | **HAMLET: ML approach to traditional ATE** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Language & Domain** | | **scores for positive label (I)** | | | **scores compared to traditional GS** | | | **scores compared to traditional GS** | | |
| | | **p** | **r** | **f1** | **p** | **r** | **f1** | **p** | **r** | **f1** |
| **en** | **corp** | 81.1 | 43.1 | 56.3 | 47.4 | 32.9 | 38.9 | 37.8 | 40.0 | 38.9 |
| | **equi** | 84.2 | 62.5 | 71.8 | 49.3 | 58.9 | 53.7 | 56.1 | 49.8 | 52.8 |
| | **htfl** | 85.9 | 70.3 | 77.3 | 51.8 | 62.6 | 56.7 | 52.9 | 36.8 | 43.4 |
| | **wind** | 75.2 | 80.2 | 77.6 | 39.7 | 63.9 | 48.9 | 38.2 | 55.5 | 45.3 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **fr** | **corp** | 85.0 | 29.1 | 43.4 | 57.0 | 25.4 | 35.2 | 42.2 | 28.2 | 33.8 |
| | **equi** | 81.0 | 39.3 | 52.9 | 45.8 | 39.7 | 42.5 | 49.6 | 43.8 | 46.5 |
| | **htfl** | 90.8 | 55.6 | 68.9 | 66.2 | 48.3 | 55.8 | 58.0 | 48.2 | 52.7 |
| | **wind** | 61.4 | 51.0 | 55.8 | 29.0 | 56.3 | 38.3 | 27.6 | 48.6 | 35.2 |
| **nl** | **corp** | 81.5 | 19.1 | 30.9 | 47.1 | 23.3 | 31.2 | 38.7 | 46.9 | 42.4 |
| | **equi** | 92.2 | 37.1 | 52.9 | 62.5 | 45.3 | 52.5 | 68.8 | 54.3 | 60.7 |
| | **htfl** | 86.7 | 67.2 | 75.7 | 59.3 | 70.1 | 64.3 | 61.2 | 50.1 | 55.1 |
| | **wind** | 62.5 | 78.0 | 69.4 | 36.3 | 71.7 | 48.2 | 33.7 | 72.7 | 46.1 |
| **Averages:** | | **80.6** | **52.7** | **61.1** | **49.3** | **49.9** | **47.2** | **47.1** | **47.9** | **46.1** |

*Table 5: Best scores (as percentages) out of 3 trials per corpus for RNN with monolingual BERT embeddings with IO labelled data, compared to results of HAMLET on the same data.*

First, the results of the RNN per corpus will be discussed. These can differ substantially per corpus. For instance, scores are consistently worst in the domain of corruption for all languages, often by a large margin. This was not surprising, as it was reportedly the most difficult corpus to annotate and clearly resulted in the lowest scores for HAMLET as well. The corpus on heart failure reaches the highest scores in all languages except for sequential scores in English. For the other two corpora, results are more mixed and depend on the type of scores: wind energy gets consistently higher sequential f1-scores than dressage, but lower f1-scores compared to the traditional GS. Similarly, the conclusions concerning the impact of the languages differs depending on which scores are consulted. Average sequential f1-scores are much higher in English (70.8% versus 55.3%. and 57.2% on average for French and Dutch), but these differences are much smaller for f1-scores compared to the traditional GS, where the averages for English, French, and Dutch are 49.6%, 43.0%, and 49.1% respectively. Lower scores for French may be due to a higher ratio of multi-word terms, which are more difficult to detect (see also section 7). Precision is also much lower for the scores compared to the traditional GS. In conclusion, while language and domain do appear to have an impact on results, there are clearly other factors that have a big impact as well and more research is required to identify these dynamics. Additionally, while both types of scores are relevant for the evaluation of sequential ATE, this analysis shows how they can have a big impact on the conclusions that are reached from the results. Therefore, as research into sequential approaches for ATE continues to evolve, reporting both types of metrics whenever possible could be a helpful best practice.

### 6.3 Sequential, Neural Approach vs. Traditional, Feature-based Approach

As described in section 4.3, HAMLET is a supervised machine learning approach to ATE according to the traditional, non-sequential methodology (extracting a list of unique CTs). For

this comparison, HAMLET was always trained and evaluated on the same corpora and GS data, and the best out of three trials was selected as well. As mentioned in section 3, the way the scores are calculated compared to the traditional GS puts the sequential system at a slight disadvantage compared to HAMLET. It is, therefore, remarkable that the sequential approach is sometimes able to outperform HAMLET (if only by a small margin) despite this disadvantage. A second observation is that the approaches clearly have different strengths. For instance, the RNN performs much better on the English heart failure corpus, but HAMLET obtains much higher scores for the Dutch corpus on corruption. Both methodologies tend to extract slightly more terms than are present in the GS. Over all corpora, there are 18,801 unique GS terms; HAMLET extracts 19,379 CTs and when sequential results of the RNN are converted to a list of unique CTs, this results in 20,194 CTs.

Investigating the results in more detail reveals more differences. For instance, the average length (in number of tokens) of CTs extracted by the RNN is 1.8, which is the same as in the GS; average length of HAMLET CTs is only 1.4. While the RNN is better at extracting longer terms, it also extracts many long false positives, with outliers of CTs up to 35 tokens. In contrast, the longest term extracted by HAMLET counts 7 tokens. These long false positives may, at least in part, be due to the choice of a binary (IO) labelling scheme which cannot always distinguish between the boundaries of terms. Analysing term frequencies revealed more interesting differences. The traditional approach to ATE is notoriously bad at extracting rare terms, which is an important disadvantage, considering domain-specific corpora will often contain many rare terms. In the ACTER dataset, 48.4% of all 18,801 unique GS terms are *hapax terms*, i.e., they occur only once. Even though HAMLET's machine learning approach to traditional ATE performs slightly better in that respect than a rule-based approach (Rigouts Terryn et al. 2019), HAMLET still only extracts 34.9% of all hapax terms, versus a total average recall of 47.9%. The difference is smaller for the RNN, where recall on hapax terms is 42.8%, versus a total average recall of 49.9%. Interestingly, both systems find slightly different hapax terms, since 58.5% of all hapax terms are found by at least one of both systems.

While all methodologies in this project operate with a binary definition of terms, the traditional dataset is more fine-grained and distinguishes between Specific Terms, Common Terms, Out-of-Domain Terms, and Named Entities (see section 3). The proportion differs per corpus, but out of all 18,801 GS terms in all corpora, 55% are Specific Terms, 27% Common Terms, 3% Out-of-Domain Terms, and 16% Named Entities. There are too few Out-of-Domain Terms to draw meaningful conclusions about those, but the results for the other three labels will be briefly discussed. Previous research on HAMLET showed that the system tends to extract a disproportionate number of Named Entities. This is also the case in our experiments, where HAMLET's total recall for all Named Entities in the GS is 63.6%, versus only 46.4% for Specific Terms and 42.2% for Common Terms. This was previously explained by the fact that Named Entities can be identified more easily based on characteristics like capitalisation, and that the results of a named entity recognition system are integrated as features. Specific and Common Terms can be more difficult, with many hapax terms among the former category and many ambiguous terms in the latter. The RNN's recall for Specific Terms, Common Terms, and Named Entities is 51.4%, 45.0%, and 54.8%. Both systems struggle most with Common Terms, likely because these terms, by definition, occur regularly in general language corpora as well.

Like HAMLET, the RNN is relatively better at extracting the few Named Entities than Specific and Common Terms, but the difference is smaller. Training and evaluating the RNN with the

same settings and data but without considering Named Entities positive instances, results in an average sequential f1-score on the positive label of 47.7%, and an f1 compared to the traditional GS of 39.8%. Especially for the sequential score, this is a considerable drop compared to the f1-scores when Named Entities are included: 13.4 percentage points less for the sequential score, 7.4 percentage points less for the f1-score compared to the traditional GS. A similar experiment was performed with HAMLET, reported in previous work (Rigouts Terryn et al., 2021), where HAMLET was also trained to consider Named Entities as negative instances. For HAMLET, leaving out Named Entities only led to a drop in f1-scores of 2.6 percentage points. So, concerning Named Entities, both the RNN and HAMLET appear to extract them with relatively higher accuracy than terms, but HAMLET appears to be better able to distinguish between terms and Named Entities than the RNN, which makes relatively more mistakes when trained to extract only terms.

To investigate how much these differences are due to the sequential approach, or to the use of word embeddings instead of other features, the same experiment was performed with the sequential, feature-based CRF. Training and evaluating the CRF system only on terms, without Named Entities, resulted in a sequential f1-score of 45.1% and an f1-score versus the traditional GS of 26.9%, which is, respectively, -11.9 and -6.7 percentage points compared to the approach including Named Entities. Since these differences are similar to the ones for the RNN, and much higher than for HAMLET, we tentatively conclude that sequential approaches may have more trouble distinguishing between terms and Named Entities, though this has to be confirmed with more extensive comparisons.

### 6.4 Complementarity of Results

As all three methodologies appear to have different strengths and weaknesses, further experiments were performed to investigate whether they are complementary. For now, we focused on a simple, pairwise, lenient or strict voting system. Results from two systems can be combined with strict voting (token or CT only kept if extracted by both systems) or lenient voting (token or CT kept if extracted by either system). When all three systems are combined, this principle is applied twice, e.g., the results of a lenient combination of both sequential systems are combined with the results from HAMLET using strict voting. Results from the combination of both sequential systems can be seen in Table 6. Sequential scores are only calculated when the HAMLET system is not included. The best out of three trials is selected, so the feature-based CRF approach achieves similar sequential results as the RNN. As can be seen, some combinations are able to reach higher scores than any of the methodologies by themselves, most notably a lenient combination of HAMLET and the RNN. Combining all systems does not lead to a higher f1-score but can still be useful to optimise either precision or recall.

| | **sequential scores** | | | **scores vs. trad. GS** | | |
|---|---|---|---|---|---|---|
| **solo performance** | **p** | **r** | **f1** | **p** | **r** | **f1** |
| RNN | 80.6 | 52.7 | 61.1 | 49.3 | 49.9 | 47.2 |

| | | | | | | |
|---|---|---|---|---|---|---|
| CRF | 67.9 | 57.0 | 60.2 | 33.5 | 40.0 | 35.2 |
| HAMLET | | | | 47.1 | 47.9 | 46.1 |
| **Combination of 2** | | | | | | |
| RNN+CRF: *lenient* | 67.5 | 71.1 | 67.6 | 35.1 | 55.6 | 41.7 |
| RNN+CRF: *strict* | 87.0 | 38.6 | 51.2 | 55.2 | 33.4 | 39.2 |
| HAMLET+RNN: *lenient* | | | | 42.2 | 67.0 | 50.5 |
| HAMLET+RNN: *strict* | | | | 68.6 | 30.8 | 40.6 |
| **Combination of 3** | | | | | | |
| HAMLET + [RNN+CRF: *lenient*]: *lenient* | | | | 33.8 | 69.3 | 44.4 |
| HAMLET + [RNN+CRF: *lenient*]: *strict* | | | | 61.8 | 34.2 | 42.6 |
| HAMLET + [RNN+CRF: *strict*]: *lenient* | | | | 44.5 | 56.8 | 48.4 |
| HAMLET + [RNN+CRF: *strict*]: *strict* | | | | 70.3 | 24.6 | 34.8 |

*Table 6: Scores (as percentages), averaged over all twelve corpora, first for 3 separate systems (best of 3 trials), then combinations of 2 and 3 systems with strict & lenient voting*

## 7 RNN Error Analysis

A first observation is that the RNN, despite having no explicit knowledge of POS patterns, nevertheless extracts CTs that follow logical patterns. False positives often have common POS patterns, for instance when the RNN adds a noun to an adjective or the reverse, e.g., in the English corpus on heart failure, tagging diagnostic procedures instead of only diagnostic, or circulating NT-proBNP instead of only NT-proBNP.

Many of the RNN's errors resemble errors humans might make as well, and they can even reveal inconsistencies in the GS. For instance, the tokens *cumulative hazard* (heart failure) were classified as terms twice, but counted as false positives since they had not been annotated. They should have been, as *cumulative hazard* is a specific medical term. Similarly, the token *propeller* (wind energy) was not included in the GS, but the RNN tagged all fourteen occurrences. Similar observations are made for false negatives, which are not always the best terms in the GS, e.g., the RNN does not tag *policies* (corruption), or *direction of movement* (dressage), which are both included in the GS, but probably not the best terms. While these are only anecdotal findings, it is promising that some errors can be interpreted more as disagreements on a subjective task, than as grave mistakes.

Of course, there are also other types of errors. As discussed, multi-word terms are challenging. Average precision, recall, and f1-scores for single-word terms are 57.5%, 51.3%, and 51.5% respectively; for multi-word terms this is only 40.2%, 46.7%, and 40.7%. The RNN is not always able to extract all individual tokens of multi-word terms correctly. This is especially true for terms that contain function words and adjectives, such as *heart failure with preserved*

*ejection fraction*, which occurs twenty-three times in the corpus. In two cases, all tokens are correctly identified as parts of terms; in fourteen cases, only *with* is excluded, and in seven cases both *with* and *preserved* are wrongly tagged as O. The French equivalent, *insuffisance cardiaque à fraction d'éjection préservée* only occurs four times and is correctly identified once. Twice, only the final adjective is neglected, and once both the *à* and the final adjective are not correctly identified.

One of the most common recurring errors concern ambiguous terms. These are terms which are only terminological in certain contexts and not in others, and often result in false negatives (*silence*). In some rare instances like the example above, ambiguous terms can lead to false positives, where they are tagged as terminological in a non-terminological context. In most cases, however, the opposite is true, and the ambiguous terms are not detected even when they are used in a terminological context. To go beyond anecdotal evidence to substantiate this claim, a small experiment was performed in which a domain expert was asked to list three types of terms in the corpus on dressage, in her native language (Dutch). To avoid the influence of multi-word terms and rare terms, only single-word terms were considered, and a minimum frequency of six was maintained. The selection was made without consulting the results, to avoid any bias. The three types of terms are listed below, including examples. For the examples, terms were selected that had similar ambiguous equivalents in English (provided as translations).

(1) Non-ambiguous terms that are relevant to the domain, but also well-known by non-experts
    e.g., *teugels* (*reins*), *draven* (*trotting*), *zadel* (*saddle*)
(2) Non-ambiguous, specialised terms, which are not part of general language
    e.g., *capriole*, (same in English), *longeren* (*longeing*), *renvers* (same in English)
(3) Very ambiguous terms with both a general meaning, and a domain-specific meaning that requires knowledge of the domain
    e.g., *verzameling* (*collection*), *hulp* (*aid*), *overgang* (*transition*)

In the first category, 10 terms were selected with a combined frequency of 342; in the other two, 20 terms, with combined frequencies of 620 and 608, respectively. Having a similar ratio of number of terms versus total frequency (average frequencies of 34.2, 31, and 30.4) was meant to limit the effect of term frequency on the results. Both macro- and micro-averaged recall were calculated for each category. In this case, macro-averaging means calculating the average recall of all unique terms, and micro-averaging means calculating the average recall for all instances (so considering term frequency). Scores are shown in Table 7 and confirm that, in this experiment, ambiguous terms do indeed obtain much lower recall scores than the other categories.

Larger-scale experiments are required to confirm this hypothesis, but the difference is substantial and serves as a powerful first indication. The same terms were analysed in HAMLET's output. HAMLET provides a confidence score for each CT (the higher the scores, the more likely HAMLET predicts the CT to be a true term). Macro-averaging these scores for each category revealed that HAMLET struggles with that same category of ambiguous terms (micro-averaging for HAMLET is not possible, since the system only extracts unique terms, not each instance of each term). In conclusion, this error analysis shows promising results for a very subjective task and identifies concrete issues for further research.

| | RNN | | HAMLET |
|---|---|---|---|
| | **macro-averaged recall** | **micro-averaged recall** | **macro-averaged confidence** |
| **(1) non-ambiguous common terms** | 75.8 | 73.4 | 61.1 |
| **(2) non-ambiguous specific terms** | 62.0 | 61.0 | 85.1 |
| **(3) very ambiguous specific terms** | 15.6 | 21.1 | 44.0 |

*Table 7: Scores (as percentages) for small samples of different types of single-word terms in the Dutch dressage corpus; scores from the best of 3 trials of the RNN and HAMLET*

# 8 Conclusion

As with many tasks in the domain of NLP, machine learning methodologies have become popular strategies for ATE. So far, most of these approaches have broadly followed the traditional approach to ATE, i.e., using a rule-based strategy to extract a list of CTs and then classifying and/or ranking these CTs based on how likely they are true terms. The next phase may be to step away from this approach and use sequential machine learning instead, where each token is classified as (part of) a term or not in the text itself, without first extracting CTs. This methodology has rarely been tried for ATE, so the goal of the current project was to investigate whether such a sequential approach is suited for ATE, and what its strengths and weaknesses are compared to the more traditional approach. Moreover, the use of word embeddings will likely only become more popular with the rise of such strategies, so instead of developing only a single sequential methodology, both a feature-based CRF approach and an embedding-based RNN approach were compared. Additionally, they were compared to a machine learning system that follows a more traditional approach and that used similar features as the former. Results showed that the RNN obtained a slightly more robust performance than the CRF overall, and that it compared favourably to the non-sequential approach. Another important finding was that the type of evaluation metric has a large impact on the scores. As has been observed in many previous studies, the presence of in-domain training data was shown to have a big effect on results as well.

A more in-depth error analysis revealed some of the strengths and weaknesses of the sequential RNN versus the traditional approach, like higher performance on rare terms. While it was shown that there is definite potential for a combination of different approaches, there are also terms that are still difficult for all methodologies, most notably ambiguous terms, which are common in general language and only acquire a specialised terminological meaning in a domain-specific context. Future research will, therefore, focus on combinations of the approaches, and the use of domain-specific embeddings in combination with general embeddings can be investigated to help extract the ambiguous terms. Another direction for future research is multilingual ATE from comparable corpora, i.e., the cross-lingual linking of equivalent terms based on non-aligned corpora. As multilingual embeddings were shown to work well for monolingual ATE in the current project, they are an interesting strategy to explore for cross-lingual experiments as well.

## 9 Funding Information

This research has been carried out as part of a PhD fellowship on the EXTRACT project, funded by the Research Foundation – Flanders.

## 10 References

Agić, Željko, and Ivan Vulić. 2019. 'JW300: A Wide-Coverage Parallel Corpus for Low-Resource Languages'. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3204–10. Florence, Italy: Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1310.

Akbik, Alan, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. 'FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP'. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 54–59. Minneapolis, USA: Association for Computational Linguistics.

Akbik, Alan, Duncan Blythe, and Roland Vollgraf. 2018. 'Contextual String Embeddings for Sequence Labeling'. In *Proceedings of the 27th International Conference on Computational Linguistics*, 1638–49. Sante Fe, New Mexico, USA: Association for Computational Linguistics.

Alami Merrouni, Zakariae, Bouchra Frikh, and Brahim Ouhbi. 2020. 'Automatic Keyphrase Extraction: A Survey and Trends'. *Journal of Intelligent Information Systems* 54 (2): 391–424. https://doi.org/10.1007/s10844-019-00558-9.

Amjadian, Ehsan, Diana Inkpen, T.Sima Paribakht, and Farahnaz Faez. 2016. 'Local-Global Vectors to Improve Unigram Terminology Extraction'. In *Proceedings of the 5th International Workshop on Computational Terminology*, 2–11. Osaka, Japan.

Amjadian, Ehsan, Diana Zaiu Inkpen, T. Sima Paribakht, and Farahnaz Faez. 2018. 'Distributed Specificity for Automatic Terminology Extraction'. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 24 (1): 23–40. https://doi.org/10.1075/term.00012.amj.

Astrakhantsev, Nikita, D. Fedorenko, and D. Yu. Turdakov. 2015. 'Methods for Automatic Term Recognition in Domain-Specific Text Collections: A Survey'. *Programming and Computer Software* 41 (6): 336–49. https://doi.org/10.1134/S036176881506002X.

Bay, Matthias, Daniel Bruneß, Miriam Herold, Christian Schulze, Michael Guckert, and Mirjam Minor. 2020. 'Term Extraction from Medical Documents Using Word Embeddings'. In *Proceedings of the 4th IEEE Conference on Machine Learning and Natural Language Processing (MNLP)*. Agadir, Morocco: IEEE Computer Society.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. 'Enriching Word Vectors with Subword Information'. *ArXiv Preprint in ArXiv:1607.04606 [Cs]*. http://arxiv.org/abs/1607.04606.

Bourigault, Didier. 1992. 'Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases'. In *Proceedings of the 14th Conference on Computational Linguistics-Volume 3*, 977–81. Nantes, France: Association for Computational Linguistics.

———. 1993. 'An Endogeneous Corpus-Based Method for Structural Noun Phrase Disambiguation'. In *Proceedings of the Sixth Conference of the European Chapter of*

*the Association for Computational Linguistics*, 81–86. Utrecht, Netherlands: Association for Computational Linguistics.

Cram, Damien, and Beatrice Daille. 2016. 'TermSuite: Terminology Extraction with Term Variant Detection'. In *Proceedings of The 54th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 13–18. Berlin, Germany: Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-4003.

Crammer, Koby, Alex Kulesza, and Mark Dredze. 2009. 'Adaptive Regularization of Weight Vectors'. *Advances in Neural Information Processing Systems* 22: 414–22. https://doi.org/10.1007/s10994-013-5327-x.

Davies, Mark. 2017. 'The New 4.3 Billion Word NOW Corpus, with 4--5 Million Words of Data Added Every Day'. In *Proceedings of the 9th International Corpus Linguistics Conference. Birmingham*. Birmingham, UK. https://www.english-corpora.org/now.

De Clercq, Orphée, Marjan Van de Kauter, Els Lefever, and Veronique Hoste. 2015. 'LT3: Applying Hybrid Terminology Extraction to Aspect-Based Sentiment Analysis'. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 719–24. Denver, Colorado: Association for Computational Linguistics. https://doi.org/10.18653/v1/S15-2122.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 'BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding'. *ArXiv:1810.04805 [Cs]*. http://arxiv.org/abs/1810.04805.

Dobrov, Boris, and Natalia Loukachevitch. 2011. 'Multiple Evidence for Term Extraction in Broad Domains'. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, 710–15. Hissar, Bulgaria: Association for Computational Linguistics.

Drouin, Patrick. 2003. 'Term Extraction Using Non-Technical Corpora as a Point of Leverage'. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 9 (1): 99–115.

Drouin, Patrick, Jean-Benoît Morel, and Marie-Claude L' Homme. 2020. 'Automatic Term Extraction from Newspaper Corpora: Making the Most of Specificity and Common Features'. In *Proceedings of the 6th International Workshop on Computational Terminology (COMPUTERM 2020)*, 1–7. Marseille, France: ELRA.

Fedorenko, Denis, Nikita Astrakhantsev, and Denis Turdakov. 2013. 'Automatic Recognition of Domain-Specific Terms: An Experimental Evaluation'. In *Proceedings of the Ninth Spring Researcher's Colloquium on Database and Information Systems*, 26:15–23. Kazan, Russia.

Goyal, Archana, Vishal Gupta, and Manish Kumar. 2018. 'Recent Named Entity Recognition and Classification Techniques: A Systematic Review'. *Computer Science Review* 29 (August): 21–43. https://doi.org/10.1016/j.cosrev.2018.06.001.

Graff, David, Ângelo Mendonça, and Denise DiPersio. 2011. 'French Gigaword Third Edition LDC2011T10'. Philadelphia, USA: Linguistic Data Consortium.

Habibi, Maryam, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. 2017. 'Deep Learning with Word Embeddings Improves Biomedical Named Entity Recognition'. *Bioinformatics* 33 (14): i37–48. https://doi.org/10.1093/bioinformatics/btx228.

Hätty, Anna, Michael Dorna, and Sabine Schulte im Walde. 2017. 'Evaluating the Reliability and Interaction of Recursively Used Feature Classes for Terminology Extraction'. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 113–21. Valencia, Spain: Association for Computational Linguistics.

Hätty, Anna, Dominik Schlechtweg, and Michael Dorna. 2020. ‘Predicting Degrees of Technicality in Automatic Terminology Extraction’. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 72883–89. olnine: Association for Computational Linguistics.

Hazem, Amir, Mérieme Bouhandi, Florian Boudin, and Béatrice Daille. 2020. ‘TermEval 2020: TALN-LS2N System for Automatic Term Extraction’. In *Proceedings of the 6th International Workshop on Computational Terminology (COMPUTERM 2020)*, 95–100. Marseille, France: European Language Resources Association.

Kageura, Kyo, and Elizabeth Marshman. 2019. ‘Terminology Extraction and Management’. In *The Routledge Handbook of Translation and Technology*, edited by O’Hagan, Minako.

Kageura, Kyo, and Bin Umino. 1996. ‘Methods of Automatic Term Recognition’. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 3 (2): 259–89.

Kauter, Marian van de, Geert Coorman, Els Lefever, Bart Desmet, Lieve Macken, and Véronique Hoste. 2013. ‘LeTs Preprocess: The Multilingual LT3 Linguistic Preprocessing Toolkit’. *Computational Linguistics in the Netherlands Journal* 3: 103–20.

Kim, J.-D., T. Ohta, Y. Tateisi, and J. Tsujii. 2003. ‘GENIA Corpus - a Semantically Annotated Corpus for Bio-Textmining’. *Bioinformatics* 19 (1): 180–82.

Kingma, Diederik P., and Jimmy Ba. 2015. ‘Adam: A Method for Stochastic Optimization’. In *Proceedings of 3rd International Conference for Learning Representations*. San Diego, CA. http://arxiv.org/abs/1412.6980.

Koutropoulou, Theoni, and Efstratios Efstratios. 2019. ‘TMG-BoBI: Generating Back-of-the-Book Indexes with the Text-to-Matrix-Generator’. In *Proceedings of the 10th International Conference on Information, Intelligence, Systems and Applications, IISA 2019*, 1–8. Patras, Greece. https://doi.org/10.1109/IISA.2019.8900745.

Kucza, Maren, Jan Niehues, Thomas Zenkel, Alex Waibel, and Sebastian Stüker. 2018. ‘Term Extraction via Neural Sequence Labeling a Comparative Evaluation of Strategies Using Recurrent Neural Networks’. In *Proceedings of Interspeech 2018, the 19th Annual Conference of the International Speech Communication Association*, 2072–76. Hyderabad, India: International Speech Communication Association. https://doi.org/10.21437/Interspeech.2018-2017.

Loshchilov, Ilya, and Frank Hutter. 2019. ‘Decoupled Weight Decay Regularization’. In *Proceedings of the Seventh International Conference on Learning Representations*. New Orleans, USA. http://arxiv.org/abs/1711.05101.

Macken, Lieve, Els Lefever, and Véronique Hoste. 2013. ‘TExSIS: Bilingual Terminology Extraction from Parallel Corpora Using Chunk-Based Alignment’. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 19 (1): 1–30.

Martin, Louis, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. ‘CamemBERT: A Tasty French Language Model’. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7203–19. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.645.

McCrae, John P, and Adrian Doyle. 2019. ‘Adapting Term Recognition to an Under-Resourced Language: The Case of Irish’. In *Proceedings of the Celtic Language Technology Workshop*, 48–57. Dublin, Ireland.

Meyers, Adam L., Yifan He, Zachary Glass, John Ortega, Shasha Liao, Angus Grieve-Smith, Ralph Grishman, and Olga Babko-Malaya. 2018. ‘The Termolator: Terminology

Recognition Based on Chunking, Statistical and Search-Based Scores'. *Frontiers in Research Metrics and Analytics* 3 (June). https://doi.org/10.3389/frma.2018.00019.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. 2013. 'Linguistic Regularities in Continuous Space Word Representations'. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746–51. Atlanta, GA, USA: Association for Computational Linguistics.

Okazaki, Naoaki. 2007. *CRFsuite: A Fast Implementation of Conditional Random Fields (CRFs)*. http://www.chokkan.org/software/crfsuite/.

Oostdijk, Nelleke, Martin Reynaert, Véronique Hoste, and Ineke Schuurman. 2013. 'The Construction of a 500-Million-Word Reference Corpus of Contemporary Written Dutch'. In *Essential Speech and Language Technology for Dutch*, edited by Peter Spyns and Jan Odijk, 219–47. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-30910-6_13.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 8024–35. Vancouver, Canada. http://arxiv.org/abs/1912.01703.

Pedregosa, Fabian, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. 'Scikit-Learn: Machine Learning in Python'. *Machine Learning in Python*, no. 12: 2825–30.

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. 'Deep Contextualized Word Representations'. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–37. New Orleans, Louisiana: Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1202.

Petrov, Slav, Dipanjan Das, and Ryan McDonald. 2012. 'A Universal Part-of-Speech Tagset'. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2089–96. Istanbul, Turkey: European Language Resources Association.

Pires, Telmo, Eva Schlinger, and Dan Garrette. 2019. 'How Multilingual Is Multilingual BERT?' In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4996–5001. Florence, Italy: Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1493.

Qasemizadeh, Behrang, and Siegfried Handschuh. 2014. 'The ACL RD-TEC: A Dataset for Benchmarking Terminology Extraction and Classification in Computational Linguistics'. In *Proceedings of COLING 2014: 4th International Workshop on Computational Terminology*, 52–63. Dublin, Ireland.

Qasemizadeh, Behrang, and Anne-Kathrin Schumann. 2016. 'The ACL RD-TEC 2.0: A Language Resource for Evaluating Term Extraction and Entity Recognition Methods'. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 1862–68. Portorož, Slovenia: European Language Resources Association.

Rigouts Terryn, Ayla, Patrick Drouin, Véronique Hoste, and Els Lefever. 2019. 'Analysing the Impact of Supervised Machine Learning on Automatic Term Extraction: HAMLET vs TermoStat'. In *Proceedings of the International Conference on Recent Advances in*

*Natural Language Processing (RANLP 2019)*, 1012–21. Varna, Bulgaria. https://doi.org/10.26615/978-954-452-056-4_117.

Rigouts Terryn, Ayla, Véronique Hoste, Patrick Drouin, and Els Lefever. 2020. 'TermEval 2020: Shared Task on Automatic Term Extraction Using the Annotated Corpora for Term Extraction Research (ACTER) Dataset'. In *Proceedings of the 6th International Workshop on Computational Terminology (COMPUTERM 2020)*, 85–94. Marseille, France: European Language Resources Association.

Rigouts Terryn, Ayla, Véronique Hoste, and Els Lefever. 2020. 'In No Uncertain Terms: A Dataset for Monolingual and Multilingual Automatic Term Extraction from Comparable Corpora'. *Language Resources and Evaluation* 54 (2): 385–418. https://doi.org/10.1007/s10579-019-09453-9.

———. 2021. 'HAMLET: Hybrid Adaptable Machine Learning Approach to Extract Terminology'. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 27 (2).

Rokas, Aivaras, Sigita Rackevičienė, and Andrius Utka. 2020. 'Automatic Extraction of Lithuanian Cybersecurity Terms Using Deep Learning Approaches'. In *Proceedings of the Ninth International Conference on Baltic Human Language Technologies*, 39–46. Kaunas, Lithuania: IOS Press.

Stenetorp, Pontus, Goran Topić, Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim, and Jun'ichi Tsujii. 2011. 'BioNLP Shared Task 2011: Supporting Resources'. In *Proceedings of BioNLP Shared Task 2011 Workshop*, 112–20. Portland, oregon: Association for Computational Linguistics.

Vintar, Spela. 2010. 'Bilingual Term Recognition Revisited'. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 16 (2): 141–58.

Vivaldi, Jorge, and Horacio Rodríguez. 2001. 'Improving Term Extraction by Combining Different Techniques'. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 7 (1): 31–48. https://doi.org/10.1075/term.7.1.04viv.

Vries, Wietse de, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. 'BERTje: A Dutch BERT Model'. *ArXiv:1912.09582*, December. http://arxiv.org/abs/1912.09582.

Wang, Rui, Wei Liu, and Chris McDonald. 2016. 'Featureless Domain-Specific Term Extraction with Minimal Labelled Data'. In *Proceedings of Australasian Language Technology Association Workshop*, 103–12. Melbourne, Australia.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, et al. 2020. 'Transformers: State-of-the-Art Natural Language Processing'. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-demos.6.

Wołk, Krzysztof, and Krzysztof Marasek. 2014. 'Building Subject-Aligned Comparable Corpora and Mining It for Truly Parallel Sentence Pairs'. *Procedia Technology* 18: 126–32. https://doi.org/10.1016/j.protcy.2014.11.024.

Yuan, Yu, Jie Gao, and Yue Zhang. 2017. 'Supervised Learning for Robust Term Extraction'. In *The Proceedings of 2017 International Conference on Asian Language Processing (IALP)*, 302–5. Singapore: IEEE. https://doi.org/10.1109/IALP.2017.8300603.

Zhang, Ziqi, Johann Petrak, and Diana Maynard. 2018. 'Adapted TextRank for Term Extraction: A Generic Method of Improving Automatic Term Extraction Algorithms'. *ACM Transactions on Knowledge Discovery from Data* 12 (5): 1–7.

**Email:**
ayla.rigoutsterryn@ugent.be
veronique.hoste@ugent.be
els.lefever@ugent.be

**Postal address (same for all):**
Groot-Brittanniëlaan 45,
9000 Gent,
BELGIUM