The ACL RD-TEC: A Dataset for Benchmarking Terminology Extraction and Classification in Computational Linguistics

Behrang Q. Zadeh[*] **and Siegfried Handschuh**[*†]
[*]Insight Centre of Data Analytics
National University of Ireland, Galway
[†]Department of Computer Science and Mathematics
University of Passau, Germany
{behrang.qasemizadeh, siegfried.handschuh}@insight-centre.org

Abstract

This paper introduces ACL RD-TEC: a dataset for evaluating the extraction and classification of terms from literature in the domain of computational linguistics. The dataset is derived from the Association for Computational Linguistics anthology reference corpus (ACL ARC). In its first release, the ACL RD-TEC consists of automatically segmented, part-of-speech-tagged ACL ARC documents, three lists of candidate terms, and more than 82,000 manually annotated terms. The annotated terms are marked as either valid or invalid, and valid terms are further classified as *technology* and *non-technology* terms. Technology terms signify methods, algorithms, and solutions in computational linguistics. The paper describes the dataset and reports the relevant statistics. We hope the step described in this paper encourages a collaborative effort towards building a full-fledged annotated corpus from the computational linguistics literature.

1 Introduction

Computational terminology (CT) embraces a set of algorithms that extract terms from domain-specific corpora and arrange them in domain-specific knowledge structures such as a vocabulary, thesaurus or ontology. Modern methods in CT often take a corpus-based, distributional approach to fulfil their tasks. These methods exploit data-centric, data-sensitive techniques for mining and organizing terms. Evaluation of these methods—as described in Vivaldi and Rodrïguez (2007) and Nazarenko and Zargayouna (2009)—is inherently a difficult task. Regardless of the employed metric and method for the performance comparison of CT algorithms, however, choosing a shared dataset consisting of a fixed set of documents—which can be accessed freely and easily—is a major step towards alleviating a number of obstacles in the evaluation process. From a mathematical perspective, changes in the document set will alter the underlying distribution of words and terms in the benchmark dataset. Consequently, this can vary the performance of methods. From perspectives that involve meaning interpretation, as described in L'Homme (2014), terms are defined against a context. This context is the representative of a specialized subject field and reflects the requirements of the intended application for the extracted terms. In an evaluation dataset, the specialized subject field is largely defined by the set of documents in this dataset. Therefore, variation in the set of documents can result in variant set of terms.

Creating datasets for benchmarking CT techniques have been addressed in several research efforts. The GENIA corpus is a well-known example of such reference datasets in bio-text mining: a corpus of 2000 abstracts from scientific publications in biological literature that is accompanied by the annotations of 100,000 terms organized in a well-defined ontology (Kim et al., 2003). The Colorado Richly Annotated Full Text Corpus (CRAFT) is another example of a bio-text mining dataset, which consists of 97 articles from the PubMed Central Open Access subset annotated with biomedical concepts such as 'mouse genes' (Bada et al., 2012). In a more recent effort, Bernier-Colborne and Drouin (2014) report on creating a corpus for the evaluation of term extraction in the domain of automotive engineering.

Proceedings of the 4th International Workshop on Computational Terminology, pages 52–63,
Dublin, Ireland, August 23 2014.

The use of these datasets for CT research and terminology extraction has one obstacle: the minimal prerequisite knowledge that is required to understand these specialized discourse and literature. This understanding of text is, perhaps, essential to enable a CT researcher to first comprehend and then describe a linguistic phenomenon. Hence, conducting research in these specialized fields requires a training for terminologists. For example, research in bio-text mining is often conducted by a team that includes experts in biology, bioinformaticians and computational linguists who have specialized training in this field. Conducting CT research in these specialized domains, therefore, may not be the first choice for computational linguists who have a keen interest and specialized knowledge in the computational analysis of languages—or want to train themselves to gain this knowledge.[1]

In this paper, we introduce the ACL RD-TEC: a Reference Dataset for Terminology Extraction and Classification in the domain of computational linguistics. The ACL RD-TEC is drawn from the ACL ARC (Bird et al., 2008). The ACL ARC is a fixed set of scholarly publications in the domain of computational linguistics. It has been developed with an aim to provide a platform for benchmarking methods of scholarly document processing.[2] We report further processes and annotations that have been carried out on the ACL ARC in order to move a step closer to a reference dataset of familiar materials for the CT research community.

Before describing the dataset, Section 2 delineates the terms that are used in this paper and gives a brief summary of computational terminology. In Section 3, we explain the automatic and manual processes performed to create the ACL RD-TEC and summarize the statistics of the current release. Finally, we conclude and describe our goals for the immediate future in Section 4.

2 Computational Terminology

Computational terminology inherits its complexity from the difficulties in the interpretation of meaning in language. In terminology, these complications are often summarized by the question 'what counts as a *term*?' The Oxford Dictionary defines a term as

> 'a word or phrase used to describe a thing or to express a concept, specially in a particular kind of language or branch of study'.

According to the International Organization for Standardization (ISO), a term is

> 'a verbal designation of a general concept in a specific subject field (ISO 1087-1(2000))'.

As stated by Cabré (2010), linguistically, terms are *lexical units* and carry a special *meaning* in particular *contexts*. A lexical unit is often considered as a *lexical form*—a single token, part of a word, a word or a combination of these—that is paired with a single meaning and serves as the basic element of a language's vocabulary. As stated by L'Homme (2014), terms are the denomination of items of knowledge, i.e. concepts.

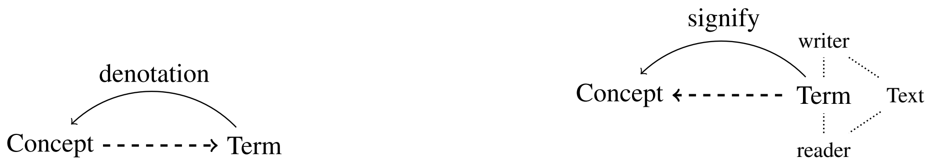
According to their lexical forms, terms are usually classified as *simple* or *complex*. Simple terms consist of one token; complex terms are composed of more than one token or word. For instance, 'lexicography' and 'multilingual terminology management' are, respectively, examples of a simple and a complex term in the domain of computational linguistics. The extracted lexical units constitute a *terminological resource*, also known as *terminology*: a specialized vocabulary of knowledge in a domain. Terms and their use are studied in a relatively young discipline, which is also called *terminology* (Cabré, 2003; Kageura, 1999):

> 'the field of activity concerned with the collection, description, processing and presentation of terms (Sager, 1990)'.

While terminology can be approached from several perspectives, e.g. as a branch of philosophy, sociology, or cognitive science, terminology is dominantly considered as a linguistic and cognitive activity.

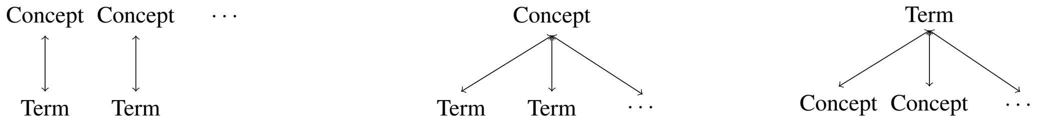
[1] Considering that knowledge and vocabulary are highly correlated, and vocabulary can be gained by exposure to literature.

[2] With an intuition similar to "eating your own dog food", as proposed in Harrison (2006).



(a) General Theory of Terminology (b) Terms as Linguistic Units

Figure 1: Association of meaning in the GTT compared to recent theories of terminology: in the GTT, terms are linguistic labels and denote concepts that exist a priori. In recent theories of terminology, e.g. CTT, however, terms are treated like other linguistic units. They signify concepts in a communicative context. In the figures above, the dashed lines indicate the direction in which the meaning of a term is elaborated according to these theories. The indicated communicative context (the dotted triangle in Figure b) can be extended in a number of ways, e.g. by considering the application of terms.



(a) One-to-One Relation (b) Synonymous Relation (c) Polysemous Relation

Figure 2: Relationships between terms and the concepts they signify: Figure 2a illustrates a mono-referential, unambiguous relationship between terms and concepts. Figure 2b shows an ambiguity that may arise when several terms denote the same concept in a synonymous relation. Figure 2c illustrates an ambiguous term-concept relation, a polysemous relationship where a term may denote several concepts.

Modern terminology is therefore pursued within a linguistic framework and as the study of specialized languages (Faber, 2012).

The meanings of terms and the process of concept denomination are studied within the framework of a 'theory of terminology'. As stated in Cabré (2003), a theory of terminology elaborates the fundamental problem of interpretation of meaning into a set of questions in which the definition of a terminological unit—and its characteristics—is the nucleus. The general theory of terminology (GTT) by Wüster (1974, as cited in Campo (2013, chap. 2)) is recognized as the first theory of terminology. The GTT, which is also known as traditional terminology, puts concepts first; terms are unambiguous linguistic labels that are defined independently of the context in which they are used (L'Homme, 2014) (Figure 1a). As implied by the given definition in ISO 1087-1(2000), the GTT is the most widely adopted theory amongst terminologists.[3] Consequently, the GTT regards terms and concepts as having mono-referential relationships (Figure 2a). The objective behind GTT, understandably, is to eliminate ambiguity in natural language to improve clarity in technical communication.

In an authoritative institutional organization that promotes or enforces standards, terms can be *made* and shared in a top-down manner; hence, the meaning of terms can be interpreted by the mechanism described in the GTT.[4] However, in practice and in many organizations, new terms are introduced in a bottom-up *synthesis* process. A lexical form (which may or may not be newly invented) in contexts that bear a concept (which may or may not be newly invented) is used frequently inasmuch as it becomes a term[5] in the organization. In practice, therefore, terms can be ambiguous: a term can refer to several concepts—similar to polysemy–homonymy in lexical semantics (Figure 2c); or, contrariwise, a particular concept can be denoted by several terms (Figure 2b). Heid and Gojun (2012) suggest that the rapid evolution of organizations as well as multi-players that are involved in an uncoordinated way, specifically in multidisciplinary domains, reinforces this situation and thus term ambiguity.

[3] Accordingly, Felber (1982) defines terminology as 'the combined action of groups of subject specialists (terminology commissions) of specialised organisations'.

[4] it is, perhaps, best demonstrated in the applications of controlled natural languages.

[5] That is, a norm.

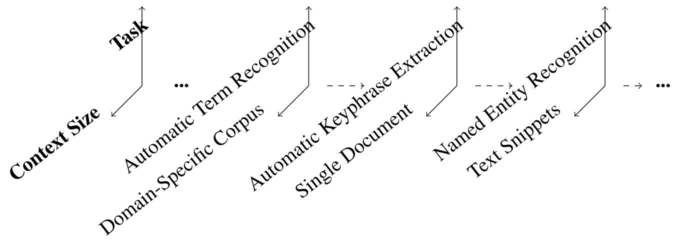


Figure 3: Lexical unit extraction tasks and the scope of the meaning: the diagram can be extended by adding new dimensions that take into the consideration characteristics of the communicative context other than the text size.

In contrast to the GTT, recent theories of terminology, e.g. the communicative theory of terminology (CTT) by Cabré (see 1999, chap. 3), acknowledges the situation stated above and takes a bottom-up distributional approach to terminology in the sense that the meanings of terms, thus the elements of domain knowledge, are not preconceived. Terms are linguistic units that are understood differently with regards to the communicative context, e.g. by the text surrounding them, the application they are used and so on. Terms signify concepts by syntagmatic and paradigmatic relations that they hold in a specialized communicative discourse (Figure 1b).[6] Methods that modern CT embraces, therefore, can be distinguished and classified by the communicative context in which they are employed.

In CT, the task of automatic term recognition (ATR) is at the centre of attention. The input of ATR is a large collection of documents, i.e. a domain-specific corpus, and the output is a terminological resource. In ATR, the meaning of the generated terms is interpreted in a wide spectrum of concepts in the domain that is being investigated and represented by the input corpus. ATR facilitates the automatic construction of terminological resources; hence, it is a significant processing resource in knowledge engineering tasks and applications such as information retrieval and machine translation.

As articulated by Kageura and Umino (1996), ATR deals with the computation of measures known as *unithood* and *termhood*. It is believed that the majority of terms in a domain are complex terms. Unithood indicates the degree to which a sequence of tokens can be combined to form a complex term. It is, thus, a measure of the *syntagmatic* relation between the constituents of complex terms: a lexical association measure to identify collocations. In the absence of explicit linguistic criteria for distinguishing complex terms from other natural language text phrases, a unithood measure construes the problem of complex term identification as the identification of *stable* lexical units (Sager, 1990).[7]

Termhood, on the other hand, 'is the degree to which a stable lexical unit is related to some domain-specific concepts' (Kageura and Umino, 1996). It characterizes a *paradigmatic* relation between lexical units—either simple or complex terms—and the communicative context that verbalizes domain-concepts. Termhood, thus, envisages the measurement of meaning. In the absence of a formal answer to the question 'what domain-specific concepts are?', devising a termhood measure for distinguishing terms and non-terms is a difficult and often conflictual task—hence, the evaluation of CT.

In ATR, the communicative context is a domain-specific corpus. ATR, therefore, should not be confused with other tasks in CT—such as keyword extraction, entity recognition, etc.—that bear a close resemblance to it. These tasks are similar to ATR in the sense that they extract stable lexical units from natural language text. However, they are different from ATR, because the meaning of the extracted lexical units, thus the termhood measure, is interpreted in a context other than a domain-specific corpus (Figure 3). For example, an automatic keyphrase extraction algorithm extracts lexical units from a single document that best describe the content of this document. Both unithood and termhood must be also measured in automatic keyphrase extraction. However, the criterion for their definition and the information available for their computation are different than ATR.

[6] As can be understood, the main difference between the GTT and the CTT is the interpretation of the process of pairing concepts and lexical forms that is mentioned in the definition of lexical units.

[7] See Evert (2004) on the application of lexical association measures for the identification of stable lexical units.

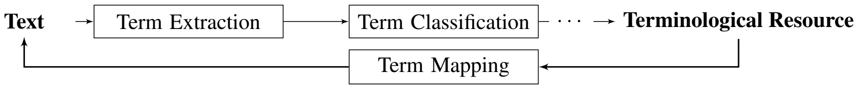


Figure 4: Significant processes in computational terminology and the direction in which they attach terms and natural language text.

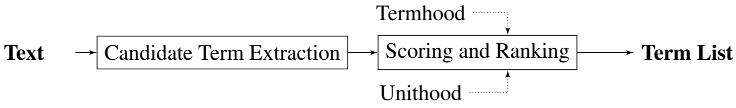


Figure 5: Prevalent architecture of the terminology extraction methods.

We can further distinguish CT methods based on the direction in which terms and text are related. Recent developments of ontological resources have stimulated a research strand that targets the reverse task of intermediary applications. The goal of these applications is to fill the gap between an available ontology, i.e. a knowledge base, and natural language text. In these tasks, given a particular concept in a knowledge base (e.g. a class and its instances in an ontology), a method—which we call *term mapping* following Krauthammer and Nenadic (2004)—decides if this concept or its instances has been mentioned in a given text snippet. Entity linking, which has been promoted through the series of Text Analysis Conferences,[8] is another term that characterizes these research efforts (see also Rao et al., 2013). The familiar task of named entity recognition (NER), as introduced at the Message Understanding Conference (Grishman and Sundheim, 1996), can also be placed in this category. In NER, the types of target terms are known prior to the extraction task, e.g. city, location and so on.

In contrast to term mapping techniques, there are methods that organize constituent terms of a terminological resource into a variety of classes. In these methods, the usage of terms in a given domain-specific corpus is assessed to decide about their membership in concept classes. If the classes are known prior to the assignment task, then the task is known as term classification; otherwise, if the classes are not known, the task is called term clustering. As suggested by Krauthammer and Nenadic (2004), these three tasks—i.e. term recognition, term classification and term mapping—are essential for automatic construction and maintenance of terminological resources and to form a closed loop between terminology and natural language text (Figure 4).

A more elaborate taxonomy of CT techniques can be obtained by distinguishing additional elements and characteristics of the communicative context, e.g. the way in which an end user benefits from the extracted terms, the role of background knowledge, linguistic characteristics of the extracted terms and so on. We leave this study for another occasion.

2.1 Prevalent Mechanism in Term Extraction Tasks

As suggested in Nakagawa (2001), the algorithms for term recognition are usually in the form of a two-step procedure: candidate term extraction followed by term scoring and ranking (Figure 5).

Candidate term extraction deals with the term formation and the extraction of candidate terms. In a few applications, candidate term extraction can assess the morphosyntactic structure of terms, e.g. as suggested in Ananiadou (1994) and Zweigenbaum and Grabar (1999), to identify candidate terms. In these methods, existing terminologies are often available before to the extraction task and employed to identify new candidate terms. Besides this, one can identify four major methods for the extraction of candidate terms: linguistic filtering based on part of speech (PoS) tag sequences, n-gram technique, linguistic filtering based on syntactic relations and techniques that rely on the presence of particular markers in text. Methods for the extraction of candidate terms are not limited to these categories. For instance, *contrastive approaches* exploit a reference corpus of general language to identify simple candidate terms (Drouin, 2004, 2003); for complex terms, the comparison between corpora is followed by one of the techniques listed above. A combination of these methods can also be employed to improve

[8] `http://www.nist.gov/tac/about/`

the results (e.g. see Aubin and Hamon, 2006).

Linguistic filters in the form of PoS tag sequence patterns are the most widely adopted technique for the extraction of candidate terms. In this method, any sequence of tokens with certain PoS tags are assumed as candidate terms (Justeson and Katz, 1995; Daille, 1995). Likewise, the knowledge about PoS patterns that *cannot* form candidate terms may be used to restrict the presence of token sequences in a list of candidate terms (Bourigault, 1992). In the n-gram technique, however, any sequences of tokens of length n, often $1 \leq n \leq 6$, that appear in the input text are considered as candidate terms. This method generates a large set of candidate terms. The number of candidate terms, therefore, is often reduced by filtering n-grams that contain tokens from a stop-word list. When linguistic processing tools are lacking, such as the case of under-resourced languages (see e.g. Pinnis et al., 2012), or the computational cost or accuracy hinders their usage, the n-gram technique is favourable.

Linguistic filters that employ syntactic relations have also been used for the extraction of candidate terms. The first sub-category of these methods use shallow parsing to identify noun phrases as candidate terms (Nakagawa, 2001). The second sub-category of these methods generate candidate terms from available terminological resources to identify term variations (Jacquemin and Tzoukermann, 1999). The third subcategory, which is often employed for multilingual term extraction, exploits the head-modifier principle to identify candidate terms (Hippisley et al., 2005). Finally, a category of candidate term extraction methods takes advantage of the presence of specific markers in input text that can be used to determine boundaries of terms, e.g. the presence of mark-up metadata in Hartmann et al. (2012).

Subsequent to candidate term extraction, a scoring procedure—which can be seen as a semantic weighting mechanism—is employed to indicate how likely it is that a candidate term is a term we would like to extract. As Figure 5 suggests, the scoring procedure usually combines termhood and unithood scores. Although several categorizations of the scoring and ranking methods can be given from a methodological point of view (e.g. statistics-based, machine learning-based, rule-based, etc.) or by the kind of information that is exploited for weighting (e.g. linguistic-based, statistical-based, hybrid) as stated earlier, all these techniques rely on the text and take a corpus-based distributional approach to score and rank terms. The usage of candidate terms in a communicative context (e.g. domain-corpus) is formulated in a machine-tractable format, e.g. in the form of a contingency table or a vector space model. It is then assessed using statistical measures, similarity metrics, language models or a set of rules, depending on the method employed and the objective of the task in hand, which defines the type of paradigmatic relation that the termhood measure characterizes.

The c-value algorithm, for instance, is an statistical method of assigning scores to candidate terms in an ATR task. It is used as a baseline in a number of ATR evaluation tasks. For each candidate term t, the c-value score of t is calculated using four criteria (Frantzi et al., 1998): the frequency of t in the corpus; the frequency of t when it appears nested in other terms longer than t; the number of those longer terms shown by T_t; and the number of the constituent words of t shown by $|t|$. The c-value score is given by

$$c\text{-}value(t) = \begin{cases} \log_2 |t| f(t) & \text{if } t \notin \text{nested} \\ \log_2 |t| \left(f(t) - \frac{1}{|T_t|} \sum_{b \in T_t} f(b)\right) & \text{otherwise} \end{cases}, \tag{1}$$

where T_t denotes the set of all the terms that contain t and are longer than t, and $f(s)$ denotes the frequency of an arbitary term s in the corpus. Other widely applied statistical measures for termhood assements in ATR include term frequency–inverse document frequency (*tf-idf*), term frequency (*tf*), and inverse document frequency (*idf*). We leave the study of scoring mechanisms for another occasion.

3 The ACL RD-TEC: Further Annotation Layers for ACL ARC

We introduce the ACL RD-TEC, a spin-off of the ACL ARC. In its first release, the ACL RD-TEC consists of manual annotations that can be used for the evaluation of ATR and term classification tasks that are explained in the previous section. The current release of the ACL ARC consists of 10,922 articles that were published between 1965 to 2006. The provided resources in the ACL ARC consist of three layers: (a) source publications in portable document format (PDF), (b) automatically extracted text from

Type	Token	Sentence	Paragraph	Section	Publication
704,085	36,729,513	1,564,430	510,366	92,935	10,922

Table 1: Summary statistics of the dataset derived from automatic processing of the ACL ARC.

PoS Tag	JJ	NN	NNP	VBG	FW	(Total)
Frequency	150	17,120	4,520	1255	2	(23,047)

Table 2: Summary statistics of the assigned PoS tags to the simple term 'parsing'. The PoS tags are from the *Penn Treebank PoS tagset*.

the articles and (c) bibliographic metadata and citation network. Each of the articles in the collection are assigned to a unique identifier that indicates the source (e.g. journal or conference) and the date (e.g. 1999, 2006, etc.) of publication.

In the preparation of ACL RD-TEC, we further employed the SectLabel module of Luong et al.'s (2010) ParsCit tool[9] for the automatic identification of logical text sections in ACL ARC's raw text files. Using a set of heuristics, sections such as 'bibliography' and 'acknowledgements' are removed from the corpus and are organized in separate files. In addition, text cleaning is performed, e.g. broken words and text segments are joined, footnotes and captions are removed and sections are organised into paragraphs. The sectioning process is followed by the text segmentation process using the OpenNLP sentence splitter[10] and the Stanford tokenizer. The text is then annotated by the Stanford PoS tagger.[11] The process is finalized by making an inverted index of the cleansed full-text documents and assigning unique identifiers to each one of the extracted linguistic units: types (i.e. PoS-tagged and lemmatized words), sentences, paragraphs and (sub)sections. All of these units are stored in separate flat tables, in which all units, except types, are presented as tuples, consisting of pairs of unique identifiers and their relative locations in the text units they constitute. Therefore, text units can be easily traced back to the publications that they appeared in. The statistics of the resulting data are given in Table 1.

Afterwards, candidate terms are extracted from the processed corpus using three methods: PoS-based filtering, n-gram-based technique and noun phrase (NP) chunking. In order to devise PoS sequence patterns and maximum length of candidate terms, we started with an observation of sample valid terms, their PoS sequences patterns and length in the corpus. We extracted 3301 sentences that contained the lemma 'technology'. We then identified 476 valid terms in these sentences, 65% of which had lengths of 2 and 3 tokens; only 5% were longer than 5 tokens.[12] Similar to the method proposed by Ittoo et al. (2010), to alleviate the problem of erroneous PoS tagging, we formulated the PoS patterns for candidate term extraction based on the actual output of the employed PoS tagger. All the occurrences of the identified terms were searched for in the corpus and all the PoS tag sequences assigned to them were extracted. Amongst the 100 extracted patterns, to keep a balance between correct and incorrect patterns resulted from erroneous PoS tagging, we chose 31 patterns P_l^i of maximum length 5 that satisfy the equation

$$\frac{f(P_l^i)}{\sum_{j:length(P^j)=l} f(P^j)} > \frac{1}{10^l}, \tag{2}$$

where f denotes the frequency of a PoS pattern and l is its length.

For example, the term 'parsing' is extracted as a valid term in this procedure. This simple term is encountered 23,047 times in the corpus. As shown in Table 2, the employed PoS tagger assigned several different PoS tags to this term. Assuming that these PoS tags are the only patterns of length 1, only *NN* and *NNP* satisfy the given formula in Equation 2 above and are added to the inventory of valid PoS patterns for terms of length $l = 1$. The rest of PoS patterns—*VBG, JJ* and *FW*—are discarded. As it can be understood from the right hand side of the equation, when the length of PoS patterns increases, the stated criteria for their selection process becomes easier (e.g. 0.01 for terms of length $l = 2$ instead of

[9] Release version 110505 (`http://aye.comp.nus.edu.sg/parsCit/`).

[10] Release version 1.5.2 (`http://opennlp.apache.org/`).

[11] Release date 9 July 2012; see Toutanova et al. (2003) for a description of the PoS tagger tagset.

[12] We eliminate definite and indefinite determiners from the terms.

Method	Total#	Length = 1	Length = 2	Length = 3	Length = 4	Length = 5
PoS-based	$1,322,445$	$271,064$	$741,448$	$284,725$	$23,384$	$1,824$
n-gram	$9,339,303$	$236,053$	$1,054,792$	$2,187,041$	$2,880,665$	$2,980,752$
NP Chunk	$1,813,222$	$142,636$	$706,051$	$623,633$	$248,505$	$92,397$

Table 3: Summary statistics of the extracted candidate terms.

0.1 for terms of length $l = 1$). The list of devised PoS patterns is included in the distributed package.

We repeated the procedure described above by extracting sentences that contain lemmas other than 'technology', e.g. 'algorithm', 'method', 'framework'and 'theory'. There is no evidence to support that the extracted patterns are specific to a category of terms (e.g. technology terms). These patterns seem to be generic enough to extract terms of any category. We support this claim based on the conducted manual verification of the extracted candidate terms. In these extracted sentences, the only terms that are longer than 5 tokens are various transliteration of the term 'very-large-vocabulary speaker-independent continuous speech recognition'. Based on these observations and the previous studies reported on the length of terms (e.g. see Maynard, 2000; Bonin et al., 2010), we believe the maximum length of 5 tokens is a fair trade-off between accuracy and recall in the process of candidate term extraction.

The sentences in the corpus are scanned for occurrences of the devised PoS patterns. Any sequences of tokens that conform to any of these patterns is considered as a candidate term. The extracted token sequences construct the list of PoS-based candidate terms. Based on the above observation, in the n-gram-based extraction of candidate terms, n is set to $1 \leq n \leq 5$ tokens. In addition, n-grams that begin with a token from a stop-word list[13] are discarded. The remaining n-grams form the second list of candidate terms. The extracted sentences from the corpus are also chunked by the OpenNLP chunker. NP chunks that are not longer than 5 tokens constitute the third list of candidate terms. As other lists of candidate terms, determiners are removed from the NP chunks. From all the above lists, we eliminate candidates that are shorter than 3 characters. Candidate terms are further augmented by their frequency in the corpus, distinct documents, sections, and paragraphs and stored separately. Table 3 shows a summary statistics of the extracted candidate terms.

In an ideal scenario, each occurrence of a candidate term in each sentence could have been annotated to identify the particular concept–class that the term signals in that context. Such annotations could have been used in all the tasks described in the previous section. In the absence of an agreed taxonomy of concepts and classes for computational linguistics and—more importantly—the required resources to carry out this complex manual annotation task, achieving the ideal goal at once and in a single step seems infeasible. In order to keep it manageable, we begin the manual annotation task by the verification of the candidate terms in vocabulary lists as is suggested in the previous evaluations of ATR algorithms.

To proceed with the annotation task, the extracted candidate terms are sorted using scores that are obtained from several ATR algorithms, e.g. the c-value score (Equation 1). The annotators are provided with an annotation guideline, consisting of basic definitions (such as the given description in the earlier sections), rules (e.g. how to deal with term variation, misspelled terms and so on) and examples.[14] During this process, the annotators are provided with a tool to access concordance view of candidate terms in the ACL ARC corpus.[15] The annotators are asked to envisage a mind map of computational linguistics topics and perceive the candidate terms in this map. For a given lexical form t in the list of candidate terms, if t refers to a significant concept in the computational linguistics domain,[16] the annotators are ask to mark t as valid. However, this does not guarantee that all the occurrences of t in the corpus are valid terms. For instance, 'natural language' is a lexical form that appears in the corpus as a term on several occasions, e.g. in

'$\cdots$ a *natural language* is a scheme of communication$\cdots$'.

[13] The SMART stop-word list built by Chris Buckley and Gerard Salton, which can be obtained from `goo.gl/rBQNbO`.

[14] The annotator guideline can be accessed in the distributed package.

[15] We used the preloaded version of the ACL ARC in the Sketch Engine Corpus Query System available at `https://the.sketchengine.co.uk/bonito/run.cgi/first_form?corpname=preloaded/aclarc_1`

[16] That is, if they can situate the term in their envisaged mind map.

	Total#	Length = 1	Length = 2	Length = 3	Length = 4	Length = 5
Technology Terms	13,832	757	8,674	3,822	538	41
Invalid Terms	61,818	15,908	33,502	11,027	1,211	170
Valid Terms	22,027	1,495	14,146	5,677	657	52
Total Annotated	83,845	17,403	47,648	16,704	1,868	222

Table 4: Summary statistics of the annotated candidate terms.

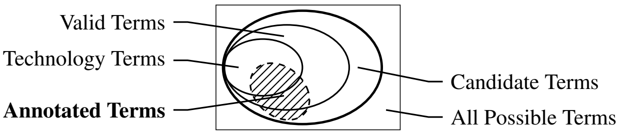


Figure 6: Relationships between candidate terms, valid terms, technology terms and annotated terms. Candidate term extraction extracts a subset of all possible terms. ATR targets the identification of valid terms amongst candidate terms. Technology terms are a subset of valid terms. The dashed area shows the set of annotated terms.

However, there are a number of occurrences of 'natural language' that cannot be considered as term, e.g.

'$\cdots$ the speech and *natural language* groups at SRI reported results $\cdots$'.

On the other hand, if t is annotated as invalid, then there must be no occurrence of t in the corpus that can be counted as a term. In the current version, 83,845 terms are annotated as either valid or invalid.

Furthermore, valid terms in the annotated list of terms are classified as those that can signal a technology concept. If 'genes' are an essential category of concepts in an ontology that characterizes biological discipline, we speculate that the presence of technology as a category of concepts is essential in any ontology or terminological resource that describes an applied discipline like computational linguistics. As to our definition, technology terms indicate concepts such as methods, algorithms and processes that are designed, developed and employed to accomplish a certain task in order to fulfil a practical purpose, i.e. to address a research problem (see also the task in Kovaevi et al., 2012). In computational linguistics, examples of these terms are 'parsing', 'information retrieval', and more delicate terms such as 'linear interpolation'.

In order to distinguish technology terms amongst other categories of terms, annotators are provided with several definitions of technology and its known examples in computational linguistics.[17] In addition, the annotators are exposed to materials on philosophy of technology, e.g. Franssen et al. (2013), and introduced to the task of 'tech mining' in Porter and Cunningham (2005). Despite all these efforts, because establishing a precise definition of technology is infeasible, classification of valid terms to technology and non-technology terms, to a great extent, relies on the intuition of experts who participated in the annotation task. The annotators are allowed to use other sources of information than the ACL ARC, e.g. web search, in order to decide about the technology class membership of valid terms. The process of annotating technology terms in the lists of extracted candidate terms is facilitated by supervised machine learning-based methods of term weighting, e.g. as reported in Zadeh and Handschuh (2014a,b). Table 4 shows the current statistics of the annotated terms. Figure 6 illustrates relationships between candidate terms, valid terms and technology terms.

Similar to the valid terms, terms that are annotated as technology terms do not exclusively belong to this class. For example, 'computational linguistics' is a lexical form that can be classified as a technology term, e.g., in

'$\cdots$promising area of application of *computational linguistics* techniques$\cdots$'.

However, it can also signal other concepts such as a scientific discipline, e.g. in

'$\cdots$theoretical work in *computational linguistics*$\cdots$'

[17] Those terms that are explicitly named as technology in literature are taken as the examples of technology terms. To make a list of examples, we identified these terms using simple patterns such as '$\cdots$ X is a technology$\cdots$'.

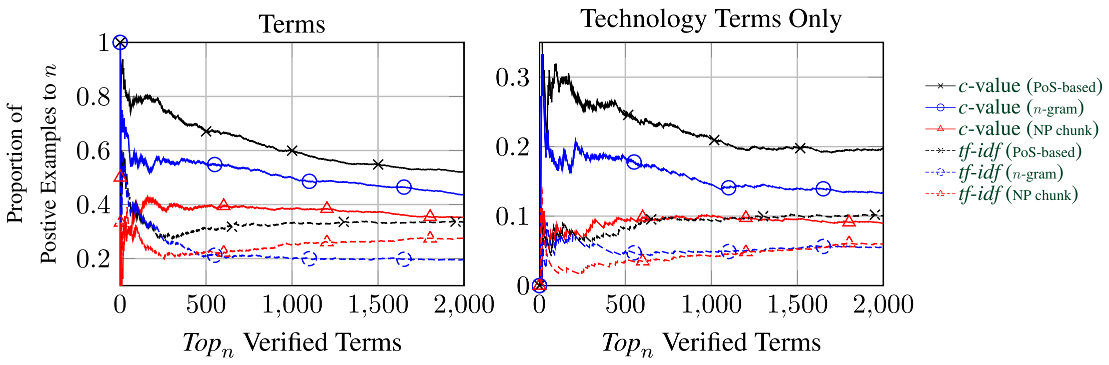


Figure 7: Comparison between c-value and tf-idf.

, as well as a community, e.g. in

> ‘· · · pursued by the *computational linguistics* community · · · ’.

The data, perhaps, speaks better for itself. Thus, we invite the interested reader to explore the annotated set of terms in order to gain more insight into the performed annotation task. The dataset can be obtained freely from the European Language Resources Association, catalogue reference ELRA-T0375.[18]

While we hope more researchers become involved in the annotation task, in the current release, all the annotations are made by one person. In order to assess the quality and reliability of the annotations, we carried out two preliminary experiments. In the first experiment, a list of terms consisting of 250 terms that have been particularly difficult to annotate are annotated by a researcher who is familiar with terminology. For example, we particularly found it hard and conflicting to annotate terms that start with words such as ‘automatic, automated, stat-of-the-art, scalable, rapid, full, fast’ and so on, e.g. in terms such as ‘fast clustering, ‘fast classification, and ‘fast prototyping. In addition, deciding on the inclusion of certain categories of terms is difficult. For example, one may consider ‘people’ and ‘organizations’ as valid domain terms, while another person—with her own specific interest and expertise—may consider these as invalid terms. This problem is more subtle about categories such as ‘languages’ and ‘linguistic units’. For instance, one may consider ‘English’ and ‘French’ as well as ‘clitic’ and ‘suffix’ as terms; however, another person may not consider them as valid terms in the domain of computational linguistics (e.g. one may found them too generic to be considered as valid terms). As to our experience, the more specific we are about the concept categories, the easier it is to annotate the terms. We made sure sample of these terms are included in the assessment of the annotations. We report an observed agreement A_o of 0.758 and Cohen'kappa coefficient κ of 0.517 for this set of terms (see Artstein and Poesio, 2008, for definition of A_o and κ).

In the second experiment, two postgrad students in the area of natural language processing were given a list of 389 terms and asked to identify technology terms. The list of annotated terms were then compared with the annotations in the dataset. The results are $A_o = 0.840$ and $\kappa = 0.655$ for the first comparison and $A_o = 0.775$ and $\kappa = 0.533$ for the second comparison. These measures over the annotations generated by the participants in the evaluation task are $A_o = 0.828$ and $\kappa = 0.627$.

As a usage example of the constructed dataset, we use the annotations for the comparison of the top n terms in the list of candidate terms that are weighted and sorted using Frantzi et al.'s (1998) c-value and term frequency–inverse document frequency (tf-idf). We hope other researchers in the domain are intrigued by the numbers reported in Figure 7 and report the performance of other algorithms.

4 Conclusion and Future Work

The saying ‘the shoemaker's son goes barefoot’ is perhaps true when it comes to the state of terminological resources that characterize computational linguistics domain. We report a small action towards

[18] http://catalog.elra.info/index.php; the annotated terms are also available from https://github.com/languagerecipes/the-acl-rd-tec.

building a terminological resource from the ACL ARC, which can be used for the evaluation of computational terminology methods. There are currently three sets of candidate terms, which are augmented by their frequency in various logical text segments in the corpus and are presented in tabulated inverse index files. More than 82,000 of these terms are annotated manually as valid and invalid, in which valid terms are further classified as technology and non-technology terms. The built resource can facilitate the evaluation of a number of methods in computational terminology. We invite other researchers to embellish the dataset by adding their own lists of candidate terms and manual annotations.

During the annotation process we have identified several frequent concepts other than technology and methods in the computational linguistics domain, e.g. grammar formalism, theories, measures, language resources, tasks and applications. We hope to continue our effort by adding annotations for at least one of these concepts. Adding a new concept class will allow us to evaluate term disambiguation methods. The application of clustering techniques for identification of term variations amongst the annotated terms and their manual annotation is another goal that can be achieved in the near future. These small steps, collectively, can provide the shoemaker's son with a fine pair of leather boots.

Acknowledgements

We thank the anonymous reviewers for their constructive comments, which helped us to improve the paper. In addition, we thank Professor Marie-Claude L'Homme for her helpful advice. We also thank Kartik Asooja, Georgeta Bordea, Sapna Negi and Bianca Pereira who participated in the inter-annotator agreement experiment. This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

2000. ISO 1087-1:2000 terminology — vocabulary — part 1: Theory and application.

Sophia Ananiadou. 1994. A methodology for automatic term recognition. In *Proceedings of the 15th conference on Computational linguistics - Volume 2*, COLING '94, pages 1034–1038.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Comput. Linguist.*, 34(4).

Sophie Aubin and Thierry Hamon. 2006. Improving term extraction with terminological resources. In *Advances in Natural Language Processing*, LNCS 4139, pages 380–387.

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William Baumgartner, K Cohen, Karin Verspoor, Judith Blake, and Lawrence Hunter. 2012. Concept annotation in the craft corpus. *BMC Bioinformatics*, 13(1):161.

Gabriel Bernier-Colborne and Patrick Drouin. 2014. Creating a test corpus for term extractors through term annotation. *Terminology*, 20:1:5073.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *LREC'08*. Marrakech, Morocco.

Francesca Bonin, Felice Dell'Orletta, Simonetta Montemagni, and Giulia Venturi. 2010. A contrastive approach to multi-word extraction from domain-specific corpora. In *LREC'10*. ELRA, Valletta, Malta.

Didier Bourigault. 1992. LEXTER: a natural language tool for terminology extraction. In *COLING '92*. pages 977–981.

M. Teresa Cabré. 1999. *TERMINOLOGY: THEORY, METHODS AND APPLICATIONS*. John Benjamins.

M. Teresa Cabré. 2003. Theories of terminology their description, prescription and explanation. *Terminology*, 9:2:163–199.

M. Teresa Cabré. 2010. *Handbook of translation studies*, volume 1, chapter Terminology and translation, pages 356–365.

M. Teresa Cabré, Anne Condamines, and Fidelia Ibekwe-SanJuan. 2005. Introduction: Application-driven terminology engineering. *Terminology*, 11:1–19(18).

Ángela Campo. 2013. *The reception of Eugen Wsters work and the development of terminology*. Ph.D. thesis, Université de Montréal.

Beatrice Daille. 1995. Combined approach for terminology extraction: lexical statistics and linguistic filtering. Technical report, UCREL, Lancaster University.

Patrick Drouin. 2003. Term extraction using non-technical corpora as a point of leverage. *Terminology*, 9(1):99–115.

Patrick Drouin. 2004. Detection of domain specific terminology using corpora comparison. In *LREC'04*.

Stefan Evert. 2004. *The Statistics of Word Cooccurrences Word Pairs and Collocations*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart.

Pamela Faber. 2012. *A cognitive linguistics view of terminology and specialized language*, volume 20, chapter Terminology and Specialized Language, pages 13–33. Walter de Gruyter.

Helmut Felber. 1982. Computerized terminology in termnet: The role of terminological data banks. *Term banks for tomorrows world: Translating and the Computer*, 4:8–20.

Maarten Franssen, Gert-Jan Lokhorst, and Ibo van de Poel. 2013. Philosophy of technology. In *The Stanford Encyclopedia of Philosophy*. Winter 2013 edition.

KaterinaT. Frantzi, Sophia Ananiadou, and Junichi Tsujii. 1998. The c-value/nc-value method of automatic recognition for multi-word terms. In *Research and Advanced Technology for Digital Libraries*, LNCS 1513, pages 585–604.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference-6: A brief history. In *Proceedings of COLING (Vol. 96).*, pages 466–471.

Warren Harrison. 2006. Eating your own dog food. *IEEE Software*, 23(3):5–7.

Silvana Hartmann, György Szarvas, and Iryna Gurevych. 2012. Mining multiword terms from wikipedia. In *Semi-Automatic Ontology Development: Processes and Resources*, pages 226–258. IGI Global, Hershey, PA, USA.

Ulrich Heid and Anita Gojun. 2012. Term candidate extraction for terminography and cat: and overview of ttc. In *Proceedings of the 15th Euralex International Congress*. Oslo, Norway.

Andrew Hippisley, David Cheng, and Khurshid Ahmad. 2005. The head-modifier principle and multilingual term extraction. *Nat. Lang. Eng.*, 11(2):129–157.

Ashwin Ittoo, Laura Maruster, Hans Wortmann, and Gosse Bouma. 2010. Textractor: A framework for extracting relevant domain concepts from irregular corporate textual datasets. In *BIS*, LNBIP 47, pages 71–82. Springer.

Christian Jacquemin and Evelyne Tzoukermann. 1999. NLP for term variant extraction: synergy between morphology, lexicon and syntax. *Natural language information retrieval*, 7:25–74.

John S. Justeson and Slava M. Katz. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Nat. Lang. Eng.*, 1.1:9–27.

Kyo Kageura. 1999. On the study of dynamics of terminology: A proposal of a theoretical framework. *Research Bulletin of the NACSIS*, 11:1–10.

Kyo Kageura and Bin Umino. 1996. Methods of automatic term recognition: A review. *Terminology*, 3.2 (1996):259–289.

J. . D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182.

Aleksandar Kovaevi, Zora Konjovi, Branko Milosavljevi, and Goran Nenadic. 2012. Mining methodologies from nlp publications: A case study in automatic terminology recognition. *Computer Speech & Language*, 26(2):105 – 126.

Michael Krauthammer and Goran Nenadic. 2004. Term identification in the biomedical literature. *J. of Biomedical Informatics*, 37(6):512–526.

Marie-Claude L'Homme. 2014. Terminologies and taxonomies. *Oxford Handbooks Online*.

Minh-Thang Luong, Thuy Dung Nguyen, and Min-Yen Kan. 2010. Logical structure recovery in scholarly articles with rich document features. *IJDLS*, 1(4):1–23.

Diana Maynard. 2000. *Term recognition using combined knowledge sources*. Ph.D. thesis, Manchester Metropolitan University.

Hiroshi Nakagawa. 2001. Automatic term recognition based on statistics of compound nouns. *Terminology*, 6.2:195–210.

Adeline Nazarenko and Haïfa Zargayouna. 2009. Evaluating term extraction. In *Proceedings of the International Conference RANLP-2009*, pages 299–304. Association for Computational Linguistics, Borovets, Bulgaria.

Mārcis Pinnis, Nikola Ljubešić, Dan Ştefănescu, Inguna Skadiņa, Marko Tadić, and Tatiana Gornostay. 2012. Term extraction, tagging, and mapping tools for under-resourced languages. In *TKE 2012*.

Alan L Porter and Scott W Cunningham. 2005. Tech mining. *Competitive Intelligence Magazine*, 8(1):30–36.

Delip Rao, Paul McNamee, and Mark Dredze. 2013. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing.

Juan C. Sager. 1990. *Practical Course in Terminology Processing*, chapter Term Formation: theory and practice, pages 61–87. John Benjamins Publishing Company.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL '03*. pages 173–180.

Jorge Vivaldi and Horacio Rodrïguez. 2007. Evaluation of terms and term extraction systems: A practical approach. *Terminology.* 13:225–248.

Eugen Wüster. 1974. Die allgemeine terminologielehre–ein grenzgebiet zwischen sprachwissenschaft, logik, ontologie, informatik und den sachwissenschaften. *Linguistics*, 12(119):61–106.

Behrang Q. Zadeh and Siegfried Handschuh. 2014a. Evaluation of technology term recognition with random indexing. *LREC'14*. ELRA, Reykjavik, Iceland.

Behrang Q. Zadeh and Siegfried Handschuh. 2014b. Investigating context parameters in technology term recognition. *COLING Workshop on Synchronic and Diachronic Approaches to Analyzing Technical Language (SADAATL)*.

Pierre Zweigenbaum and Natalia Grabar. 1999. Automatic acquisition of morphological knowledge for medical language processing. In *Artificial Intelligence in Medicine*, LNCS 1620, pages 416–420.