Unsupervised Training Set Generation for Automatic Acquisition of Technical Terminology in Patents

Alex Judea[1] Hinrich Schütze[2] Sören Brügmann[3]
[1]Heidelberg Institute for Theoretical Studies, Heidelberg, Germany
[2]Center for Information and Language Processing, University of Munich, Germany
[3]Brügmann Software GmbH, Papenburg, Germany

Abstract

NLP methods for automatic information access to rich technological knowledge sources like patents are of great value. One important resource for accessing this knowledge is the technical terminology of the patent domain. In this paper, we address the problem of automatic terminology acquisition (ATA), i.e., the problem of automatically identifying all technical terms in a document. We analyze technical terminology in patents and define the concept of technical term based on the analysis. We present a novel method for labeling large amounts of high-quality training data for ATA in an unsupervised fashion. We train two ATA methods on this training data, a term candidate classifier and a conditional random field (CRF), and investigate the utility of different types of features. Finally, we show that our method of automatically generating training data is effective and the two ATA methods successfully generalize, considerably increasing recall while preserving high precision relative to a state-of-the-art baseline.

1 Introduction

A large part of our technological knowledge is encoded in patents. Methods for automatically finding information in patents and inferring information from patents are thus of great value. An important step in getting access to patent information is identification of technical terminology, i.e., finding the linguistic expressions that denote the technical concepts of a patent: the methods, processes, substances and objects that are part of the invention or modified by it. In the example "The present invention relates to a **charging apparatus** of a **bicycle dynamo**", the bolded compound nouns are the main content words and refer to specific technological concepts. We call such linguistic expressions *(technical) terms* or TERMS and their totality the *(technical) terminology* of a document or domain.

We address the task of *automatic terminology acquisition* (ATA), the task of finding technical TERMS in texts without reliance on existing resources that list TERMS of the domain. In contrast to this stands *automatic terminology recognition* (ATR), which we define as finding *known* TERMS and their variants (Jacquemin and Bourigault, 2003). ATA provides input to downstream components like automatic summarization, machine translation, ontology building, information extraction and retrieval. TERMS extracted by ATA can be semantically classified or mapped to entries in a semantic database (Krauthammer and Nenadic, 2004), but we focus on identifying them without further classification in this paper.

Our main contributions are as follows. (i) We present a method for automatically labeling large amounts of training data for ATA. (ii) We show that two types of statistical classifiers trained on this training data beat a state-of-the-art baseline, indicating that the automatic labeling is of high quality. (iii) We study different feature types for ATA and investigate how much they contribute to good performance. We investigate a semi-supervised setting in which features are selected based on a manually labeled evaluation set and a completely unsupervised setting where the feature selection is performed on an automatically produced set. (iv) Finally, we show that performance strongly depends on correct identification of the boundaries of TERMS and could be enhanced considerably by improving candidate identification.

The paper is organized as follows. Section 2 gives a definition of technical terminology and provides a brief analysis of TERMS in patents. Section 3 presents related work. Section 4 describes the architecture of our ATA system: preprocessing, linguistic filtering, automatic labeling of training data, feature selection and postprocessing. Section 5 reports evaluation results and analyzes selected features and errors. Section 6 presents our conclusions.

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 290–300, Dublin, Ireland, August 23-29 2014.

2 Problem Description

Let $w_{1...k}$ be a sequence of words $w_1, w_2, \ldots, w_k$ and w_k a head noun. $w_{1...k}$ is a TERM of domain D iff (i) the head noun w_k is unmodified ($k = 1$) or (for $k > 1$) is modified by sequences of other nouns ("disk controller"), adjectives ("secondary controller") or present participles ("writing controller") and (ii) it denotes a concept specific to D.

(i) and (ii) describe the syntactic and semantic properties of a TERM, respectively. Part (i) restricts TERMS to parts of noun phrases. This is a reasonable restriction that covers most technical terms (Daille et al., 1996) and it has been frequently made in the computational terminology literature. We exclude comparatives and superlatives as modifying adjectives because they are rarely used attributively in patents and usually modify quantities or qualities of TERMS(e.g., "*higher* shunt currents"); in other words, only "positive" (base-form) adjectives are included in our definition. Note that the number of tokens per TERM is not restricted by the definition. Our approach aims to find TERMS of arbitrary length.

Part (ii) of the definition restricts TERMS to be specific to a domain D. We can set D to a general domain like 'electricity' and be on a par with many prior definitions (Ananiadou, 1994; Georgantopoulos and Piperidis, 2000; Zhang and Fang, 2010), but we can also set D to a narrow domain like 'emergency protective circuit arrangements' (IPC code[1] H02H).

Here, we choose the most general technical domain possible: the domain of all technical subjects. This is a good setting for many downstream tasks, e.g., information retrieval should benefit from a broad coverage of D. It also makes annotation easier: Non-experts can carry it out with good agreement (Section 5.1) because they simply look for all technical expressions.

The syntactic and semantic parts of our definition of TERM correspond to the concepts of *unithood* and *termhood*, respectively. Unithood is the degree to which a sequence of tokens is a linguistic unit; and termhood the degree to which a linguistic unit is a TERM of a domain (Kageura and Umino, 1996). Both aspects have to be covered by ATA systems.

Terms in Patents

In addition to traditional TERMS like simple nouns (1, "voltage"), modified nouns (2, "secondary arm") and nouns modified by prepositional phrases (3, "trajectory of the lever"), patents provide also coordinations (4, "constant and variable current") and complex constructions (5, "storage device storing a target temperature value which a battery is intended to reach").

For ATA, it seems advisable to exclude infrequent and complex nominal expressions from the definition of TERM, both from a terminological and a computational point of view. Most nominal expressions that are generally viewed as terms are single nouns, compound nouns, and nouns with an adjectival modifier (Daille et al., 1996); our syntactic definition covers these three types. Nominal expressions like (5) tend to be long; if we were to count such cases as TERMS, then it would be unclear where the TERM ends. When analyzing (5), our first take might be that there is a nucleus ("storage device") which is modified by a verbal phrase ("storing a target temperature value") and that the rest of the phrase is not part of the TERM. But it turns out that the *whole phrase* appears multiple times in its patent; it is a stable way of denoting a part of the invention. However, the underlying concept is also denoted by simpler constructions like the nucleus itself, or synonymous TERMS like "control circuit"; these simpler constructions are covered by our definition.

Coordinations like (4) mix multiple concepts (here, "constant current" and "variable current") without making this explicit on the surface. It is difficult to identify "constant current" as a potential TERM because it is non-contiguous and is only indicated by an adjective. Our treatment of coordinations in this paper is to only consider sequences satisfying the syntactic definition (i.e., "variable current") to be TERMS and discard other parts (i.e., "constant"). Of course, if both conjuncts are complete TERMS and satisfy the syntactic definition, both will be identified as TERMS.

Finally, prepositional phrases like (3) are rather infrequent compared to terms covered by our syntactic definition. They also tend to be highly ambiguous and the underlying concept is often expressed by terms covered by our definition ("lever trajectory").

3 Related Work

Previous work on ATA either employs *filtering* or *sequence models*. Filtering combines linguistic and statistical criteria for (i) *extracting a list of candidates* (typically word n-grams) based on simple linguistic criteria, (ii) *computing candidate statistics* and (iii) using ranking, classification or some other mechanism for *producing a pruned list of* TERMS *as output*. Because variation of the surface form of TERMS is limited,

[1] `wwwcms10.wipo.int/classifications/ipc/en/`

it makes sense to use word n-grams as the basis for candidate identification – even though there are cases that cannot be found this way, e.g., "constant current" or alternations like "pressure regulating valve" vs. "valve regulating pressure".

The main difference between ATA methods that rely on filtering is in how they accomplish the ranking/pruning of the candidate list. See Kageura and Umino (1996), Jacquemin (2001) and Pazienza et al. (2005) for an overview. In this paper, we accomplish this by training a statistical model to classify TERM candidates. We also run experiments with a sequence model. Our main innovation is that these models are trained on automatically labeled training data.

It is difficult to directly compare computational terminology systems because of differences in domain, language, application and task definition. As an example consider Takeuchi and Collier (2005) who report an F_1 of .742. However, their task definition includes assigning terms to pre-defined categories such as DNA and protein as opposed to simply identifying TERMS. In addition, terminologies in the biomedical and technological domains are different. In biomedicine, categories like DNA and protein dominate. For these TERMS, shape features are informative – in contrast to TERMS in patents. Another difference is that TERMS in patents tend to be long whereas DNA and proteins are often single-token abbreviations.

3.1 Training Data Collection

One of our main contributions is unsupervised training data generation (Section 4.3). Prior work has used automatically recognized training data for computational terminology, specifically for ATR (Craven and Kumlien, 1999; Hatzivassiloglou et al., 2001; Morgan et al., 2003; Zhang et al., 2010) in the biomedical domain. Given large precompiled TERM lists they search for occurrences of list elements, e.g., genes, in texts and use the occurrences they find as training examples. This is similar to *distant supervision* (Mintz et al., 2009) which also uses pre-existing resources such as gazetteers for, e.g., relation extraction.

In contrast, our method is applied to ATA for the technological domain and does not rely on precompiled resources – we make use of figure references, which are an inherent part of patents. Our method can be characterized as training data *identification*: we exploit given conditions in patents for our search of training data. In contrast, training data *recognition* methods need precompiled resources as input and search for instances of resource elements in texts.

3.2 Learning Algorithms and Features

Different learning algorithms and feature sets have been used for computational terminology. Foo and Merkel (2010) use Ripper (Cohen, 1995) with a variety of features to classify uni- and bigram TERM candidates. Hatzivassiloglou et al. (2001) compare C4.5 (Quinlan, 1993) and Naive Bayes (Duda and Hart, 1973). Zhang et al. (2010) acquire novel TERMS using CRFs and syntactic features. Takeuchi and Collier (2005) find that more training data results in higher F scores. Large training sets have the same positive effect in our experiments. Our approach has the added advantage that the training sets are generated completely automatically.

4 Approach

As discussed in the introduction, we address the problem of ATA. We use the abbreviation ATAS (automatic terminology acquisition system) to refer to our approach in general as well as to the specific implementation we evaluate in this paper.

ATAS consists of three parts: (i) training set generation, (ii) parameter selection and training of the TERM candidate classifier (ATAS-TC) and the CRF (ATAS-CRF) and (iii) identification of terminology in documents.

Processing in step (iii) is document by document because some of our features are document-based. ATAS takes a document as input and identifies all TERMS in the document, using the TERM candidate classifier or the CRF learned in (ii).

The TERM candidate classifier (ATAS-TC) decides on entire (multi-token) candidates while the CRF decides on single tokens. ATAS-TC heavily relies on candidate computation and its decisions are mutually independent, which is clearly incorrect. In contrast, ATAS-CRF is less dependent on candidate computation and models dependence of decisions correctly; but it lacks the more 'global' view of ATAS-TC on entire candidates. We want to investigate which approach is more suited for ATA.

In what follows we describe how we preprocess patents, the linguistic filters used to implement our syntactic definition of TERM, automatic labeling of training data (step (i) of ATAS), training of TERM candidate classifier and CRF (step (ii) of ATAS), features and feature selection.

4.1 Preprocessing

The preprocessing pipeline consists of the ANNIE tokenizer, OpenNLP sentence splitter, Mate POS tagger (Bohnet (2010), retrained for patents) and Mate lemmatizer. Preprocessing has a big influence on computational terminology because special domain text poses problems for off-the-shelf components. For example, patents tend to use common language words in rare functions or meanings, e.g., "said" as a de facto determiner in contexts such as "the structure of said component". Other problems are the use of special language words, e.g., substances like "triphenylphosphine" and acronyms like "AC". Such properties pose serious problems to POS taggers. Patent citations, acronyms and even product names can include punctuation, confusing sentence splitters. Chemical formulas may confuse tokenizers.

We adapted our POS tagger and sentence splitter for patent language to deal with unusual punctuation and POS tags – especially unusual POS tags of common-language words like "said". This adaptation involves training on a manually labeled training set of patent text and some other adjustments; e.g., we only allow the tag NN for the acronyms "AC", "DC" and "A/D".

4.2 Filter

We now describe how we find TERM candidates that satisfy the syntactic definition; recall that only (possibly modified) nouns can be TERMS (Section 2).

In general, candidate identification strategies using linguistic knowledge perform better. There are two different strategies of this type: (i) parsing the sentence, extracting nominal chunks from the parse and further processing the nominal chunks and (ii) POS tagging the sentence and extracting word sequences that satisfy a set of predefined POS patterns. Because many patent sentences are long and difficult to parse, we adopt the POS pattern approach in this paper. To this end, we define two simple POS-based rules for finding term candidates.[2]

PREMODS. This rule defines a modifier *sequence*. It matches a sequence of noun pre-modifiers: (JJ|"/"|VBG|RB|N(N|P))*.[3] We include RB because the POS tagger sometimes misclassifies JJ as RB. We include "/" because the tokenizer splits abbreviations containing it.

CANDIDATE. This rule defines a TERM candidate. It matches either a single noun or PREMODS followed by a noun: (PREMODS N(N|P)). The last noun must be longer than two characters. We add a flag indicating if the candidate comes before a figure reference. A figure reference consists of an optional keyword (e.g., "Figure", "Fig.") and a sequence of numbers and letters, optionally enclosed in parentheses.

We select the longest match in case of overlapping matches and the first longest match in case of overlapping matches of the same length.

These simple rules will find all TERMS – as well as many non-terms that we will train ATAS to identify – with two exceptions. First, due to POS errors some candidates are spurious. Second, unwanted modifiers may be part of candidates. E.g., the rules will only identify "same battery" as a candidate and not "battery". But only "battery" is a valid TERM. To address the latter, we manually compiled a stop list of 67 modifiers, mostly numerals ("first") and adjectives in anaphoric function ("above-mentioned"). These modifiers are removed from TERM candidates.

4.3 Automatic Labeling of Training Data

We view ATA as either a binary classification task where a TERM candidate classifier decides if a candidate is a TERM or not, or as a sequence labeling task where a CRF decides if a token (word) belongs to a TERM or not.

Large training sets are needed to train such models. Usually, these sets are produced by expensive human labeling. We present a method for generating high quality training data in an unsupervised way without the necessity of precompiled resources. In principle, our method can be used for any language for which machine-readable patents are available.

Our starting point is that patents typically contain figure references, i.e., pointers to drawings illustrating the invention or its parts. Consider the example: "...so that first **clamp-holding secondary arms** (1) ..." Here, the figure reference ("(1)") points to the illustration "Figure 1" and is preceded by the illustrated TERM ("clamp-holding secondary arms"). Illustrated TERMS may be concrete, as in this example, or abstract, e.g., a diagram illustrating properties of a method.

We call a TERM candidate that precedes a figure reference a *basic figure reference term candidate* (bFRTC). In a manual inspection of bFRTCs in 12 patents we found that almost 95% of bFRTCs were

[2] JJ, VBG, and RB are POS tags for positive adjectives, gerunds/present participles, and adverbs, respectively.
[3] '*' is the Kleene star, '?' denotes optionality, and '|' denotes alternation.

TERMS. Thus, bFRTCs can be used as positive training examples because they usually denote technical concepts; they have the advantage of being identifiable with high precision using simple patterns.

Once the bFRTCs have been identified, there is a simple way to further increase the size of the training data: we add all *extended FRTCs* (eFRTCs) to the training set, where we define an eFRTC as a TERM candidate whose suffix is a bFRTC. E.g., if we have identified "shunt current" as a bFRTC, then "AC shunt current" is an eFRTC. eFRTCs typically are hyponyms since the modifiers added at the beginning restrict the bFRTC to a more specific meaning. This kind of hyponymy is a special case of term derivation, a modification where a base term is further specified by prefixes (Daille et al., 1996). The strategy of identifying eFRTCs can also be applied to free word order languages because figure references tend to have a local and fixed occurrence pattern similar to English. We use the term FRTC to refer to both bFRTCs and eFRTCs.

We identify all FRTCs and add them as positive examples to the training set. We also add the 5% most frequent candidates as positive examples; most of them are FRTCs, so that this step usually adds few new training examples.

We label the following candidates as negative training examples: candidates appearing only once in a patent; patent citations; and measurements. Citations and measurements ("3 cm") are clear non-terms. We identify them using regular expressions. Many singletons are non-terms because they denote common language (i.e., nontechnical) concepts, e.g., "time". These heuristics for finding negative training examples are not applied to a candidate if it has the same head as a positive training example.

We exclude from the training set candidates that do not satisfy any positive or negative criteria.

4.4 Classifiers

We use the L2-regularized logistic regression of LIBLINEAR (Fan et al., 2008) as our TERM candidate classifier. We use LIBLINEAR's default normalization for continuous-valued attributes (normalization to range $[0, 1]$) and the default representation for categorical attributes. As LIBLINEAR cannot handle missing values, we replace them with their means and modes. We set the regularization parameter $c = 1$. Our sequence model is CRF++[4], order 1, with default parameters. The CRF features are adapted from the ATAS-TC features, e.g., TERM-level features (e.g., TFIDF) are propagated down to the individual tokens of the TERM. We also include word trigrams. We discretize numeric features to three values.

4.5 Features

We developed a set of 74 features for ATA. Some of these features are taken from the literature, some are specific to our approach and make use of the concept of FRTC and some exploit other properties of patents (e.g., the importance of the title and the claims in patents). A final group consists of other novel features that we designed in the course of developing our system. We now provide an overview. c refers to a TERM candidate.

Corpus and document statistics. This feature type captures termhood and unithood of c as well as the position of c's first occurrence in the document. We use a corpus of technical text C_T and a general language corpus C_G. For every $c \in C_T$ we collect the number of patents it appears in, its frequency and its FRTC frequency, i.e., the number of its occurrences that are FRTCs. Features that are intended to indicate termhood include simple frequencies and distributional characteristics (in C_T or in a single patent). Finally, we define a measure of frequency deviation (or 'keywordness') of $h(c)$, the head of c:

$$\text{bias}(h(c)) = \frac{f_{C_G}(h(c))}{|C_G|}|C_T| - f_{C_T}(h(c))$$

f_{C_G} (resp., f_{C_T}) are the frequencies in C_G (resp., C_T), $|X|$ is the sum of frequencies of all $x \in X$. $\text{bias}(h(c))$ measures the deviation between expected frequency of the head of c (estimated on C_G) and its actual frequency. The intuition here is that the frequency of a general language noun like "time" will be similar over text types, resulting in a lower bias.

Context. This feature type captures unigrams and bigrams adjacent to c as well as their POS tags.

Part-of-speech. This feature type captures the POS sequence of c.

A patent usually focuses on a narrow technological subdomain. As a result, many of its TERMS are semantically related to each other. We would like to include features that directly capture semantic similarity to other TERMS because a candidate that is semantically similar to several other already recognized TERMS is likely to be a TERM itself.

Our goal in this paper is to address ATA using simple and efficient methods. For this reason, we approximate semantic similarity using string similarity because a subset of semantically similar terms are

[4] `crfpp.googlecode.com`

	$\mathcal{T}^{\text{u}}_{\text{tdg}}$	$\mathcal{T}^{\text{l}}_{\text{test}}$	$\mathcal{T}^{\text{l}}_{\text{dev}}$	$\mathcal{T}^{\text{u}}_{\text{sel}}$
patents	365	5	11	25
word tokens	3,422,131	50,007	74,000	152,715
word types	292,994	3711	7391	4141
bFRTCs	119,316	1264	2558	6503
FRTCs	240,240	2371	4942	10,110
candidates	353,238	8836	13,099	27,164
TERMS		3814	7220	

Table 1: Data set statistics

	P	R	F_1	description
1	.704	.797	.748	mean string similarity of c and FRTCs
2	.712	.832	.767	frequency of c as an FRTC in C_T
3	.694	.887	.779	TFIDF of c
4	.703	.888	.784	is c uppercase?
5	.708	.893	.790	is c followed by a figure reference?
6	.710	.896	.792	TFIDF of $h(c)$
7	.711	.895	.793	frequency of $h(c)$ as an FRTC in C_T
8	.718	.892	.795	bias($h(c)$)
9	.720	.891	.797	# sentences with FRTCs that c occurs in
10	.720	.893	.797	C-value of c
11	.721	.893	.798	frequency of $h(c)$ in C_G

Table 2: Features selected on $\mathcal{T}^{\text{l}}_{\text{dev}}$ (setting S). c: TERM candidate. $h(c)$: head of c

also similar on the surface. E.g., the semantic similarity between "AC power supply source" and "AC supply source" also manifests itself as string similarity.

String similarity. When designing a similarity measure, we wanted it to satisfy the following criteria: (i) more words in common should result in higher scores and (ii) words in common *towards the end* of the two strings should be weighted higher than words in common at the beginning. The motivation for (ii) is that candidates differing only in initial modifiers are often cohyponyms and highly related; conversely, candidates with different heads are often not related.

To implement this, we represent a candidate c as a vector $\vec{c}$ in $|V|$-dimensional space where V is the vocabulary. $\vec{c}_i$ is set to the position of word w_i in c if it occurs and 0 otherwise. The string similarity between c and c' is then defined as the cosine of $\vec{c}$ and $\vec{c'}$. Example: for "AC power supply source" and "AC supply source", we get the vectors $(1, 2, 3, 4)$ and $(1, 0, 2, 3)$ and the cosine .927; comparing the first string with "AC power supply" with the vector $(1, 2, 3, 0)$ we get the cosine .683.

Features in our initial set of 74 that make use of this semantic similarity are: maximum similarity of c to any FRTC, average similarity of c to all FRTCs in the patent and similarity of c to the rightmost TERM candidate in the title.

Frantzi and Ananiadou (1997) define *C-value(c)* as:

$$\text{C-value}(c) = \log_2 |c|(f(c) - \frac{1}{|T_c|}\sum_{b \in T_c} f(b))$$

where T_c is the set of TERM candidates containing c and f is frequency in C_T. C-value is high for TERM candidates that are frequent and occur as parts of many other TERM candidates – this is a good indicator of termhood.

5 Experiments and Evaluation

5.1 Data Sets

We hired three students with a bachelor degree in computer science to annotate 16 patents. The test set $\mathcal{T}^{\text{l}}_{\text{test}}$ consists of 5 patents annotated by all three students. We used majority voting to produce the final gold annotations. The devset $\mathcal{T}^{\text{l}}_{\text{dev}}$ consists of the remaining 11 patents. Each $\mathcal{T}^{\text{l}}_{\text{dev}}$ sentence was annotated by one student.

Inter-annotator agreement on $\mathcal{T}^{\text{l}}_{\text{test}}$ was .76 (Fleiss' κ). Most disagreements concern *modifiers* or *common nouns* (e.g., the TERM "battery" was often not annotated). More extensive training of the annotators should reduce these problems considerably.

As unlabeled data we randomly selected 390 technology patents. We use 365 as $\mathcal{T}^{\text{u}}_{\text{tdg}}$ for training data generation and 25 as $\mathcal{T}^{\text{u}}_{\text{sel}}$ for unsupervised feature selection. We made sure the 390 documents are not

in $\mathcal{T}^{\mathrm{l}}_{\mathrm{dev}}$ and $\mathcal{T}^{\mathrm{l}}_{\mathrm{test}}$. We excluded chemical patents because standard preprocessing components often fail for chemical formulas. Table 1 gives data set statistics.

As our technical corpus C_T we use $\mathcal{T}^{\mathrm{u}}_{\mathrm{tdg}}$ and as our general corpus C_G all nouns in the 2000 most frequent English words from Project Gutenberg[5]. This list contains many general nouns which also appear in patents (e.g., "time") without containing many technical terms (e.g., "battery"); this way, C_T and C_G give us a good contrast between technical and non-technical vocabularies (cf. Section 4.5).

One obstacle to comparing systems for ATA in the technical domain is the lack of publicly available evaluation benchmarks. We are making our data sets and the annotation guidelines available[6].

5.2 Baselines

We define the *FRTC baseline* as the system that labels all FRTCs and only FRTCs as TERMS. Almost all FRTCs are TERMS, but many TERMS are not FRTCs; thus, the FRTC baseline has high precision and low recall. Our goal is to preserve high precision while considerably increasing recall, or to generalize well from FRTCs to other TERMS.

Our state of the art baseline is Z-CRF, a reimplementation of the CRF described in (Zhang et al., 2010). Its feature representation includes POS tags, unigrams, bigrams and syntactic information, e.g., the number of times a particular token is used in a syntactic function like subject in the training set. Syntactic information is extracted with Mate (Bohnet, 2010). Z-CRF is trained on $\mathcal{T}^{\mathrm{u}}_{\mathrm{tdg}}$, just as ATAS.

Our last baseline is the well-known C-value (Frantzi and Ananiadou, 1997). Like our first baseline, it needs no training data. In contrast to our first baseline, it was specifically designed for terminology acquisition. It combines observations about statistical and linguistic properties of TERMS, i.e., a candidate is preferred as a term if it is long and frequently appears as substring of other candidates. Following Frantzi and Ananiadou (1997) we regard a candidate as TERM if its C-value it not zero; unlike them, we do not restrict the length of TERMS because the computation of long terms did not pose computational problems for us.

5.3 Evaluation Setup

We evaluate ATAS using precision, recall and F_1. Evaluation is based on candidate tokens (as opposed to candidate types or word tokens); e.g., each instance of a candidate TERM that is incorrectly classified as a TERM is a false positive. Evaluation is strict in the sense that a TERM is counted as a false positive if there is a single token that is added or missed.

We evaluate ATAS in two settings. In the system (S) setting, the ATAS pipeline described in Section 4 (ATAS-TC or ATAS-CRF) is used to identify TERM candidates. This is the real-world setting since errors in TERM candidate identification – misplaced boundaries, missing candidates, etc. – are a major source of error in ATA.

We would also like to evaluate candidate classification on *gold boundaries* (manually verified boundaries of TERM candidates); this allows us to quantify by how much performance can be improved if candidate identification is perfect. However, since gold boundary annotation is expensive, we instead approximated it: (i) We run automatic TERM candidate identification. (ii) We remove all TERM candidates that overlap with gold (manually annotated) TERMS. (iii) The set of gold TERM candidates is then the union of all remaining automatically identified candidates and the manually annotated TERMS.

In the gold boundary (G) setting, we provide these gold TERM candidates to the ATAS pipeline. This allows us to evaluate the performance of TERM/non-term classification separately from TERM candidate identification.

5.4 Feature Selection

For our feature set of 74, we perform forward feature selection for the TERM candidate classifier by selecting the feature in each step that maximizes system F_1. We perform feature selection (i) on the manually labeled set $\mathcal{T}^{\mathrm{l}}_{\mathrm{dev}}$ (to gauge performance for an optimal or close-to-optimal feature set) and (ii) on the automatically labeled set $\mathcal{T}^{\mathrm{u}}_{\mathrm{sel}}$ (to gauge the performance in a completely unsupervised setting). In the following we explain both settings in more detail.

Table 2 gives the features selected in **supervised feature selection**, i.e., when features are optimized on $\mathcal{T}^{\mathrm{l}}_{\mathrm{dev}}$. Precision remains stable, except for a drop on line 3. Recall rises steadily from .797 to .893. F_1 increases from .748 to .798.

The best feature (line 1) is the mean string similarity of a TERM candidate c to all FRTCs in a document (Section 4.5). Together with the next best feature (frequency of c as an FRTC in C_T) and feature 5 (is

[5] en.wiktionary.org/wiki/Wiktionary:Frequency_lists/PG/2006/04/1-10000

[6] h-its.org/english/research/nlp/download/terminology.php

			ATAS-TC				ATAS-CRF				Baselines					
			S-SEL		U-SEL		S-SEL		U-SEL		Z-CRF		C-value		FRTC	
			S	G	S	G	S	G	S	G	S	G	S	G	S	G
$\mathcal{T}^{l}_{dev}$	1	P	.721	.838	.690	.796	.732	.844	.727	.854	.867	.891	.384	.749	.839	1.000
	2	R	.893	.892	.825	.818	.815	.699	.755	.679	.563	.607	.292	.355	.344	.353
	3	F_1	.798	.864	.752	.807	.771	.765	.741	.756	.683*	.722*	.314*	.471*	.488*	.522*
$\mathcal{T}^{l}_{test}$	4	P	.696	.753	.627	.692	.774	.832	.664	.745	.813	.840	.388	.726	.864	1.000
	5	R	.850	.853	.764	.764	.791	.743	.644	.625	.516	.559	.320	.410	.286	.302
	6	F_1	.765	.800	.689	.728	.783	.785	.654	.680	.631*	.674*	.350*	.519*	.430*	.465*

Table 3: System (S) and gold boundary (G) results with supervised (S-SEL) and unsupervised (U-SEL) feature selection. *: significantly lower than corresponding ATAS-TC and ATAS-CRF scores.

c followed by a figure reference?) this supports our intuition for using FRTCs for automatic training set generation because they are indeed strong indicators for termness. Additionally, feature 9 indicates that candidates occurring often with FRTCs in sentences are probably TERMS. Feature 4 (is c uppercase?) is selected because uppercase TERM candidates are often abbreviations and TERMS.

Feature 3 (TFIDF of c) hurts precision, but increases recall, resulting in increased F_1. This feature models the hypothesis that a TERM is frequent in some patents but does not occur in many patents. Patent writers often invent novel TERMS rather than using standard ones to make finding a patent hard. Thus, a TERM candidate that occurs often in a few patents could be such an obfuscating TERM.

TFIDF is low for TERMS with small term frequency. Features 6 (TFIDF of $h(c)$) and 10 (C-value of c) can help correctly identify such TERM candidates as TERMS.

Features 8 and 11 incorporate information from the general purpose corpus C_G. Feature 8 contrasts the frequency of c in C_G with its frequency in C_T – frequencies of TERMS are higher in C_T, frequencies of non-terms are similar in both corpora. Feature 11 is complementary to this. It makes it more probable that c is a non-term if its head appears more often in C_G. Additionally, string similarity with the patent's title is an effective feature.

Unsupervised feature selection, i.e., selection on $\mathcal{T}^{u}_{sel}$, selected seven features that are similar to those selected by supervised selection and that we will discuss now. The best unsupervised feature (*maximum* string similarity, 1) and the best supervised feature (*mean* string similarity) both capture partial string overlap of c and FRTCs. For similar reasons, the feature "string similarity of c and rightmost NP in patent title" (2) – which exploits the importance of the title in analogy to the importance of figure references – is selected.

Other selected features (relative patent frequency of c and its head (3, 4), number of patent sentences in which c occurs with FRTCs (5), patent frequency of $c = 1?(6)$) are also similar to the features selected in the supervised setting. They capture frequency distributions of c. However, while many features in the supervised setting capture distributions of c in C_T, in the unsupervised setting, distributions of c in the patent are more important. The reason may be that C_T-based features (which use all technical text as opposed to the relevant patent in question) are harder to recognize as good predictors if the set used for selection is automatically labeled and hence noisier.

The last unsupervised feature captures the length of c in tokens (7). Manual inspection revealed that on average TERMS have more tokens than non-terms (1.9 vs. 1.3).

5.5 ATAS Results

Table 3 gives evaluation results for ATA on $\mathcal{T}^{l}_{dev}$ and $\mathcal{T}^{l}_{test}$. We report results for the ATAS versions (ATAS-TC, ATAS-CRF) and for the baselines (Z-CRF, C-value, FRTC) as well as for using supervised (S-SEL) and unsupervised feature selection (U-SEL) in system setting (S) and gold boundary setting (G).

Differences in F_1 between ATAS and baselines (marked with a †) are significant at $p < .01$.[7] If not stated otherwise, numbers below are for the system setting (S).

We note that F_1 of the ATAS versions is consistently and considerably better than all baselines in all settings. E.g., line 6 shows system F_1 on $\mathcal{T}^{l}_{test}$ of ATAS-TC (.765 for S-SEL, .689 for U-SEL) and ATAS-CRF (.783 for S-SEL, .654 for U-SEL) compared to Z-CRF (.631), FRTC (.430), and the C-value baseline (.350) . The better results mainly come from higher recall (except for C-value, which is also beaten in precision). In general, precision of the baselines is higher, but recall much smaller than for ATAS. This shows that (i) statistical classifiers can be successfully trained for ATA using our method

[7]We use approximate randomization (Yeh, 2000) for all significance tests in this paper.

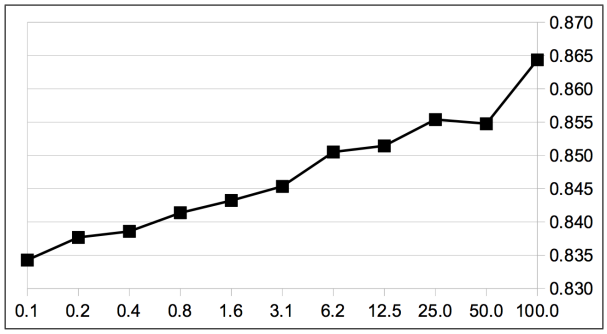

Figure 1: System F_1 as a function of training set size (in percent) in setting G.

for automatically generating training data and (ii) these classifiers beat a state-of-the-art system in both S-SEL and U-SEL settings.

Comparing S-SEL and U-SEL shows that precision and recall for U-SEL are lower than for S-SEL. For instance, F_1 of ATAS-TC on $\mathcal{T}^{\text{l}}_{\text{test}}$ is .765 for S-SEL and .689 for U-SEL; F_1 of ATAS-CRF is .783 for S-SEL and .654 for U-SEL (line 6). In general, we note a bigger drop in recall than in precision, indicating that U-SEL does not generalize as well as S-SEL. However, the U-SEL numbers are significantly better than the Z-CRF FRTC, and C-value baselines.

When comparing ATAS-TC with ATAS-CRF we note that ATAS-CRF consistently has higher precision and lower recall. In most cases, ATAS-TC has considerably higher recall, leading to higher F_1. This is not surprising given that feature selection was performed for ATAS-TC. Nevertheless, ATAS-CRF can compete with ATAS-TC in terms of F_1. Furthermore, ATAS-CRF produces more stable results because it shows less variance in F_1 across settings.

Comparing S and G scores shows that knowing exact boundaries has a great impact on results, especially on precision; looking at S-SEL numbers in line 4 in Table 3, precision for ATAS-TC (resp., ATAS-CRF) is .696 in S vs. .753 in G (resp., .774 in S vs. .832 in G). Similar differences also hold for U-SEL numbers. In general, ATAS-TC profits more from knowing exact boundaries than ATAS-CRF. This leads us to the conclusion that the linguistic filter would greatly benefit from a (statistical) measure of unithood. Note that this also holds for the baselines; deciding about the termness of gold boundary candidates seems to be easier, especially for C-value.

All observations hold for $\mathcal{T}^{\text{l}}_{\text{dev}}$ and $\mathcal{T}^{\text{l}}_{\text{test}}$. However, numbers are higher for $\mathcal{T}^{\text{l}}_{\text{dev}}$ because the ratio of FRTCs to candidates is higher than for $\mathcal{T}^{\text{l}}_{\text{test}}$ (38% vs. 27%) which improves classification performance on $\mathcal{T}^{\text{l}}_{\text{dev}}$ – this holds for ATAS as well as for the baselines.

To investigate the quality of the extracted training data, consider Figure 1. It shows F_1 in setting G as a function of training set size in percent of the total training set $\mathcal{T}^{\text{u}}_{\text{tdg}}$. For each evaluation point, we randomly add training examples from the full set. F_1 starts at .834 for 0.1% of training data (344 training examples) and rises to .864 for 100% (353,238 examples), with a small drop at 50%. Note that 1000 examples roughly correspond to one annotated patent. The main results of this experiment are that (i) a modest amount of automatically labeled training data gives good performance and (ii) the more automatically labeled data the better. The last point is not a trivial finding, given that training data was generated automatically. The logarithmic graph shows a nearly linear increase in F_1 for each doubling of the training data.

To further investigate the quality of the generated training data, we compared automatically and manually produced training examples. We compare results for 13238 manual and 13238 automatic labels (setting G, ATAS-TC). We get precision and recall of .811 and .805 for manual and .762 and .850 for automatic annotations, resulting in similar F1 scores: .808 vs. .804 for manual and automatic annotations, respectively. We believe that the differences in recall are an artifact of the randomization we performed before removing automatic training samples. Manual labels are entire patents; in contrast, automatic labels come from all patents in the training set, leaving us with a more diverse set than the manual version.

5.6 Error Analysis

We found two major types of false negatives. First, infrequent TERMS are problematic. It is hard to judge termness when having limited information about a candidate, especially if it appears only once or twice in a document. Second, POS errors prevent the system from finding some candidates; e.g., the noun "current" is frequently mistagged as adjective. Incorrect POS tags also lead to incorrect boundaries.

We found four major types of false positives. First, incorrect modifiers lead to partially incorrect TERMS. 27% of false positives are of this type. Second, incorrectly recognized figure references cause incorrect system decisions; e.g., our patterns incorrectly parse an expression like "value *PBA*" as a figure reference even though it is instead a named output of a component. Third, very frequent non-terms are commonly classified as TERMS. Almost all frequent candidates are TERMS, so that the TERM candidate classifier has difficulty correctly identifying the exceptions from this pattern.

Finally, if a candidate is a TERM in one context it may be a non-term in another. A good example for this are general single token TERMS like "apparatus". Before figure references they are TERMS, e.g., "one preferred form of apparatus 22". In such cases the figure reference serves as a disambiguator. However, in other positions they are non-terms, e.g., "They include braces, collars, splints and other similar apparatus".

6 Conclusion and Future Work

This paper introduces a method for ATA with two novel aspects: (i) new powerful features for ATA and (ii) a procedure for generating an ATA training set in an unsupervised fashion. The training set generation method produces high quality training data, even when compared to manual annotations. It is language-independent: It can be applied to patents in any language if the definition of TERM candidates is modified for the target language. It is also domain-independent: it can be applied to patents of any domain. The training data can be successfully used to train ATA models, both TERM candidate classification as well as CRF models. Even in a completely unsupervised setting the models outperform a state-of-the-art baseline. We found that using more automatically labeled training data and using better TERM boundaries results in better performance.

In future work, we plan to incorporate TERM variation patterns (Daille et al., 1996; Jacquemin, 2001) in the expansion process to decrease the number of FNs and increase recall. We would also like to improve the terminology identification module because we found that incorrect identified boundaries affect performance greatly.

Finally, we are planning to extend our approach to languages other than English. Our methods are language-independent to the extent that a body of patents exists for many common languages. Since we generate the training set automatically, all we need to do to cover another language is to adapt the linguistic filters for candidate identification.

Acknowledgments. This work was supported by the European Union (Project Topas, FP7-SME-2011 286639) and by SPP 1335 *Scalable Visual Analytics* of Deutsche Forschungsgemeinschaft (DFG grant SCHU 2246/8-2). We would like to thank the anonymous reviewers for their helpful comments and suggestions, and Bianca and Luca for their support.

References

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.

Bernd Bohnet. 2010. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97, Beijing, China, August.

William W. Cohen. 1995. Fast effective rule induction. In *Twelft International Conference on Machine Learning (ML95)*, pages 115–123.

Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In Thomas Lengauer, Reinhard Schneider, Peer Bork, Douglas L. Brutlag, Janice I. Glasgow, Hans-Werner Mewes, and Ralf Zimmer, editors, *ISMB*, pages 77–86. AAAI.

Béatrice Daille, Benoît Habert, Christian Jacquemin, and Jean Royauté. 1996. Empirical Observation of Term Variations and Principles for their Description. *Terminology*, 3(2):197–258.

Richard O. Duda and Peter E. Hart. 1973. *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc, 1 edition.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Jody Foo and Magnus Merkel. 2010. Using machine learning to perform automatic term recognition. In *Proceedings of the LREC 2010 Workshop on Methods for automatic acquisition of Language Resources and their Evaluation Methods*, pages 49–54.

Katerina T. Frantzi and Sophia Ananiadou. 1997. Automatic Term Recognition using Contextual Cues. In *Proceedings of 3rd DELOS Workshop*, Zurich, Switzerland.

Byron Georgantopoulos and Stelios Piperidis. 2000. Term-based Identification of sentences for Text Summarisation. In *Proceedings of Second International Conference on Language Resources and Evaluation (LREC2000)*, pages 1067–1070, Athens, Greece.

Vasileios Hatzivassiloglou, Pablo Ariel Dubou, and Andrey Rzhetsky. 2001. Disambiguating proteins, genes, and rna in text: a machine learning approach. In *ISMB (Supplement of Bioinformatics)*, pages 97–106.

Christian Jacquemin and Didier Bourigault. 2003. Term extraction and automatic indexing. In Ruslan Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, chapter 33. Oxford University Press.

Christian Jacquemin. 2001. *Spotting and Discovering Terms Through Natural Language Processing*. MIT Press, April.

Kyo Kageura and Bin Umino. 1996. Methods of automatic term recognition: A review. *Terminology*, 3(2):259–289.

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

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Alex Morgan, Lynette Hirschman, Alexander Yeh, and Marc Colosimo. 2003. Gene Name Extraction Using FlyBase Resources. In *Proceedings of ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 1–8, Sapporo, Japan.

Maria Teresa Pazienza, Marco Pennacchiotti, Michele Vindigni, and Fabio Massimo Zanzotto. 2005. Ai/nlp technologies applied to spacecraft mission design. In *Proceedings of the 18th international conference on Innovations in Applied Artificial Intelligence*, IEA/AIE'2005, pages 239–248, London, UK, UK.

John Ross Quinlan. 1993. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Koichi Takeuchi and Nigel Collier. 2005. Bio-medical entity extraction using support vector machines. *Artificial Intelligence in Medicine*, 33(2):125–137, February.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics - Volume 2*, COLING '00, pages 947–953, Stroudsburg, PA, USA.

Xing Zhang and Alex Chengyu Fang. 2010. An ATE system based on probabilistic relations between terms and syntactic functions. In *10th International Conference on Statistical Analysis of Textual Data*, pages 1135–1143, Sapienza, Italy, June.

Xing Zhang, Yan Song, and Alex Chengyu Fang. 2010. How well conditional random fields can be used in novel term recognition. In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, pages 583–592, Tohoku University, Sendai, Japan, November.