frontiers in Research Metrics and Analytics

ORIGINAL RESEARCH
published: 15 June 2018
doi: 10.3389/frma.2018.00019

The Termolator: Terminology Recognition Based on Chunking, Statistical and Search-Based Scores

Adam L. Meyers[1]*, *Yifan He*[1], *Zachary Glass*[1], *John Ortega*[1], *Shasha Liao*[2], *Angus Grieve-Smith*[3], *Ralph Grishman*[1] and *Olga Babko-Malaya*[4]

[1] Department of Computer Science, New York University, New York, NY, United States, [2] Google Inc., Mountain View, CA, United States, [3] Department of Information Technology, Columbia University, New York, NY, United States, [4] BAE Systems, Burlington, MA, United States

The Termolator is an open-source high-performing terminology extraction system, available on Github. The Termolator combines several different approaches to get superior coverage and precision. The in-line term component identifies potential instances of terminology using a chunking procedure, similar to noun group chunking, but favoring chunks that contain out-of-vocabulary words, nominalizations, technical adjectives, and other specialized word classes. The distributional component ranks such term chunks according to several metrics including: (a) a set of metrics that favors term chunks that are relatively more frequent in a "foreground" corpus about a single topic than they are in a "background" or multi-topic corpus; (b) a well-formedness score based on linguistic features; and (c) a relevance score which measures how often terms appear in articles and patents in a Yahoo web search. We analyse the contributions made by each of these components and show that all modules contribute to the system's performance, both in terms of the number and quality of terms identified. This paper expands upon previous publications about this research and includes descriptions of some of the improvements made since its initial release. This study also includes a comparison with another terminology extraction system available on-line, Termostat (Drouin, 2003). We found that the systems get comparable results when applied to small amounts of data: about 50% precision for a single foreground file (*Einstein's Theory of Relativity*). However, when running the system with 500 patent files as foreground, Termolator performed significantly better than Termostat. For 500 refrigeration patents, Termolator got 70% precision vs. Termostat's 52%. For 500 semiconductor patents, Termolator got 79% precision vs. Termostat's 51%.

Keywords: terminology extraction, terminology, technology forecasting, information extraction, multiword expressions

OPEN ACCESS

Edited by:
Philipp Mayr, Leibniz Institut für Sozialwissenschaften (GESIS), Germany

Reviewed by:
Chengzhi Zhang, Nanjing University of Science and Technology, China
Isola Ajiferuke, University of Western Ontario, Canada

***Correspondence:**
Adam L. Meyers
meyers@cs.nyu.edu

Received: *09 February 2018*
Accepted: *15 May 2018*
Published: *15 June 2018*

Citation:
Meyers AL, He Y, Glass Z, Ortega J, Liao S, Grieve-Smith A, Grishman R and Babko-Malaya O (2018) The Termolator: Terminology Recognition Based on Chunking, Statistical and Search-Based Scores. Front. Res. Metr. Anal. 3:19. doi: 10.3389/frma.2018.00019

INTRODUCTION

Automatic terminology extraction systems aim to collect word sequences to be used as Information Retrieval key words, terms to be included in domain-specific glossaries or ontologies. Terms are also tracked by technology forecasting applications and are potential arguments of information extraction relations. Terminology extraction systems such as the ones described in Damerau (1993), Drouin (2003), Navigli and Velardi (2004), and others find terminology by comparing the distribution of potential terms in foreground and background corpora, where a foreground corpus consists of text that is about some topic of interest and a background corpus consists of varied documents about all different topics. Potential terms being considered can be single words, bigrams, other n-grams or a constituent type such as a noun groups (Justeson and Katz, 1995).

This paper describes ***the Termolator***, an open source terminology extraction system available on Github[1]. We build on our previous Termolator papers (Meyers et al., 2014a, 2015), adding subsequent improvements (caching information for efficiency, an improved stemming procedure) and additional evaluation experiments, including a comparison to Termostat, another terminology extraction program (Drouin, 2003). The Termolator selects the terms (scientific noun sequences) that are characteristic of a particular technical area. The system identifies all potential instances of terminology in sets of files using a sequential pattern matching process called chunking. Our chunker is similar to the noun group chunkers used in many natural language processing systems, but includes additional constraints so that the selected noun group chunks must contain words belonging to specialized vocabulary classes including: out-of-vocabulary words, nominalizations, technical adjectives, and others. To find chunks that are characteristic of a topic, the system compares the frequencies of particular terms in 2 sets of documents: the foreground corpus (documents about a single topic) and the background corpus (documents about a mixture of topics). It uses several statistical measures to make this determination including Document Relevance Document Consensus or DRDC (Navigli and Velardi, 2004), Term Frequency-Inverse Document Frequency (TFIDF, Spärck Jones, 1972) and Kullback-Leibler Divergence or KLD (Cover and Thomas, 1991; Hisamitsu et al., 1999). For each foreground set of documents, the system produces a list of terms, which is initially ordered based on the distributional means just described. Two other types of scores are factored in to the system's ranking: a well-formedness score based on linguistic constraints, and a relevance score, based on how often a Yahoo (https://search.yahoo.com) web-search results for that term point to patents or articles. The final ranking is used to extract the top terms. We have found that given about 5000 foreground documents and 5,000 background documents, we can generate about 5,000 terms that are approximately 80–85% correct. The system has been tested on US patents, Web of Science abstracts, Open American National Corpus documents (http://www.anc.org/data/oanc/), books from project Gutenberg (https://www.gutenberg.org/) and English journal articles from the PubMed Central corpus (http://www.ncbi.nlm.nih.gov/pmc/). We have implemented some of these components of a Chinese version of the system and are considering developing a system for Spanish for future work. Many other terminology extraction systems, mentioned throughout this paper, also compare the distribution of potential terms in a foreground corpus with a background in order to select characteristic terms. The main things that make Termoloator different are: our particular chunking method for selecting potentential terms (other systems use single words, n-grams or standard noun groups); and our reranking (or filtering methods). Thus Termolator combines the advantages of knowledge-based and statistical techniques to produce superior results.

[1]Termolator's NYU website: http://nlp.cs.nyu.edu/termolator/ English System: https://github.com/AdamMeyers/The_Termolator/ Chinese System: https://github.com/ivanhe/termolator/

SYSTEM DESCRIPTION (ENGLISH)

System Overview

As depicted in **Figure 1**, Termolator runs in three stages: (1) terminological chunking and abbreviation; (2) distributional ranking; and (3) filtering (or reordering). The first stage identifies instances of potential terms in text. The second stage orders the terms according to their relative distribution in the foreground and background corpora. The final stage reorders the top N terms from the second stage based on a well-formedness metric and a relevance metric[2]. The so-called filtering criteria sometimes simply rule-out terms completely, and other times they change their ranking in the term list[3]. The assumption behind the ranking is that the higher ranked terms are preferred over lower ranked ones in three respects: (1) higher ranked terms are less likely to be errors (ill-formed as noun groups) and less likely to be "normal" noun sequences, phrases that are part of the general vocabulary, rather than specialized vocabulary (aka terminology); (2) higher ranking terms tend to be more characteristic of a particular field of interest than lower ranking terms; and (3) higher ranking terms tend to have greater relevance than the low ranking ones, i.e., specialists and others are currently more interested in the concepts represented by the high ranking terms.

Stage 1: Terminological Chunking and Abbreviation

In this section, we describe the component of our system designed for identifying terms in sentences, independent of their distribution in sets of documents. Like Justeson and Katz (1995), we assume that most instances of terminology are noun

[2]There are actually two parameters to determine the cutoff of the terms considered for the third stage. There is a top N parameter (which defaults to 30,000) and a top P percent parameter (which defaults to 30% of the initial term list). P% of the entire list is considered unless it exceeds N terms, in which case we just use N terms. Our defaults assume that the lowest 70% of a ranked list of terms are likely to be of low quality. At the same time, for our purposes we rarely need to look at more than 30 K terms.

[3]For example, a score of zero in any of the metrics will cause the term to simply be ruled out, whereas a higher ranking may cause it to be more preferred or less preferred.

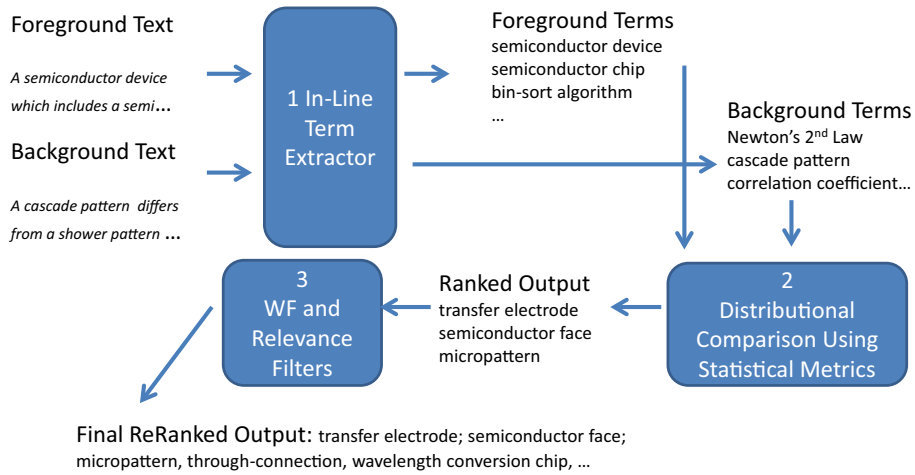

FIGURE 1 | Termolator system overview.

groups, head nouns and pre-modifiers other than determiners. Consequently, we currently exclude non-noun instances of terminology (verbs like *calcify* or *coactivate*; adjectives like *covalent* or *model-theoretic* and adverbs like *deterministically* or *stochastically*). Unlike previous approaches, we consider only a subset of noun groups as we adapt a more stringent set of chunking rules than used for standard noun group detection. We also identify an additional set of terms by means of rules for identifying abbreviations. We call these terms in-line terms, as this stage is geared toward finding instances of term tokens in documents, rather than identifying classes of terms (types) across a set of documents (the larger task of the full-system)[4].

Terminology Chunking

We incorporate into our chunking rules requirements that constituents contain nominalizations, out of vocabulary words, technical adjectives and other classes of a more fine-grained nature than typical parts of speech used in noun chunking. Nominalizations, such as *amplification* and *radiation* are identified and classified using the NOMLEX_PLUS dictionary (Macleod et al., 1998; Meyers et al., 2004)[5], contributing to the ranking of the terms *optical amplification medium fiber* and *optical radiation*. Out of vocabulary words (e.g., *photoconductor* and *collimate*) are words not found in the lexicon COMLEX Syntax (Macleod et al., 1997), thus selecting terms like *electrophotographic photoconductor* and *optical collimate*[6].

Technical adjectives are adjectives found in COMLEX or classified by a POS tagger that end in *-ic, -cal*, or *–ous,* but are not part of a manually selected out-list (e.g., *public*, *jealous*)[7]. The chunking component is modeled as a finite state machine (FSM) using a fine-grained set of parts of speech (FPOS) to determine transitions between **Beginning**, **Ending**, **Inside,** and **Outside** states in the style of Ramshaw and Marcus (1995). These noun chunks are sequences of these categories. The rules omit preceding determiners, normal adjectives and other words that are not likely to be parts of instances of terminology[8]. The FSM identifies potential terms (PTs). PTs that meet an additional set of constraints are marked as in-line terms. The FSM uses the following FPOS tags:

- **Adjectives**, words with POS tags JJ, JJR or JJS, are subdivided into:
  - **TECH-ADJ**: If an adjective ends in a suffix indicating (*-ic, -cous, -xous*, and several others) it is a technical word, but it is not found in our list of exceptions, it is marked TECH-ADJ.
  - **NAT-ADJ**: An adjective, usually capitalized, that is the adjectival form of a country, state, city or continent, e.g., *European, Indian, Peruvian*, ...
  - **CAP-ADJ**: Adjective with the first letter capitalized (but not NAT-ADJ).
  - **ADJ**: Other adjectives

[4]We identify small number of additional term types, specifically chemical formulas and gene sequences, using regular expressions.

[5]NOMLEX-PLUS is described in Meyers et al. (2004). It extends the original Nomlex lexicon described in Macleod et al. (1998).

[6]We have found the word list in COMLEX to be a reasonably good filter for identifying in-vocabulary words. For some domains, we have had to supplement with dictionaries of special in-vocabulary words, words that we treat as out-of-vocabulary, even though they are in COMLEX. For example, we have a dictionary of chemical names, which we always use. We also have a legal dictionary, which we are experimenting with for the legal domain (e.g., court decisions). If extended to social media, we of course would have to add additional dictionaries as well. For the most part, however, mostly words that don't occur in COMLEX tend to be genuine neologisms. The "basic" lexicon of the language actually changes very slowly.

[7]There are 1,445 adjectives in COMLEX with these endings, so it was possible to quickly go through these by eye in a few hours. All but 237 of these adjectives were deemed to be technical.

[8]This set of constraints is based on informal observations of the composition of valid terms in corpora. We validate this set of constraints by showing that results that are constrained this way have higher scores than results that are not so constrained, as discussed below in the Evaluation section.

- **Nouns** are marked NN or NNS by the POS tagger and are the default POS for out of vocabulary (OOV) words. POS tags like NNP, NNPS, and FW (proper nouns and foreign nouns) are not reliable for our POS tagger (trained on news) when applied to patents and technical articles. So NOUN is also assumed for these. Subclasses include:
  - **O-NOUN**: (Singular or plural) nouns not found in any of our dictionaries (COMLEX Syntax plus some person names) or nouns found in lists of specialized vocabulary which currently include chemical names.
  - **PER-NOUN**: Nouns beginning with a capital that are in our dictionary of first and last names.
  - **C-NOUN**: Nouns with POS NN that are not marked O-NOUN or PER-NOUN. A subset of these are nominalizations, a distinction used by constraints applied to the output of the FSM.
  - **PLUR-NOUN**: Nouns with POS NNS nouns that are not marked O-NOUN or PER-NOUN. These include plurals of nominalizations.
- **Verbs that can be modifiers:**
  - **ING-VER**B—verbs marked VBG. These verbs ending in –ing can function as head nouns and can pre-modify nouns.
  - **EN-VERB**—verbs marked VBN and VBD. Past-participles can pre-modify nouns like adjectives. Although these are normally marked VBN, we assume that VBD is a common POS tagging error when past tense and past participles share the same form of a given verb (e.g., *cooked* can be either VBN or VBD).
- **POSS**: Part of speech of the 's, separated from a possessive noun by the POS tagger.
- **PREP**: All prepositions (POS IN and TO)
- **ROM-NUM**: Roman numerals (I, II, ..., MMM)
- **Other**: The tag used for all other parts of speech, including verbs hat are neither ING-VERBs not EN-VERBS.

The transitions in the FST are represented in **Table 1** The states are: **B-T** (**Beginning of Term**); **I-T** (**Inside Term), E-T (End of Term)**, **O** (**Outside term), S (Start Sentence), and E (End Sentence)**. This finite state machine recognizes potential terms (PTs). A **PT** is a sequence consisting of 1 **B-T**, followed by 0 or more **I-T** and an optional **E-T**. This can be represented by the following context free phrase structure rule:

$$\textbf{Potential Term} \rightarrow \textbf{B} - \textbf{T I} - \textbf{T}^{*}\textbf{E} - \textbf{T?} \quad (1)$$

where the Kleene star (*) means 0 or more instances and the question mark indicates optionality. As per **Table 1**. each transition to a new state is conditioned on combinations of previous FPOS, current FPOS and the previous state. For example, the table suggests that if (i) the previous word is an out of vocabulary noun (O-noun), a common singular noun (C-NOUN) or plural noun (PLUR-NOUN; (ii) the current FPOS is a roman numeral (ROM-NUM); and (iii) the previous chunk tag is either B-T or I-T, then the new chunk tag should be E-T, a transition which could help identify a term like ***GFP-myosin II***.

The PTs recognized by the FSM are filtered out unless they meet several constraints. To be accepted by the system, an in-line **term** must meet all of the following criteria:

1. It must contain at least one noun.
2. It must be more than one character long, not counting a final period.
3. It must contain at least one word consisting completely of alphabetic characters.
4. It must not end in a common abbreviation from a list (e.g., cf., etc., . . .).
5. It must not contain a word that violates a morphological filter, designed to rule out numeric identifiers (patent numbers), mathematical formulas and other non-words. This rules out tokens beginning with numbers that include letters; tokens including plus signs, ampersands, subscripts, superscripts; and tokens containing no alphanumeric characters at all, etc.
6. It must not contain any word from a list of common patent section headings.

Additionally, each in-line term **T** must satisfy at least one of the following conditions:

1. T contains at least one O-NOUN.
2. T consists of at least 4 words, at least 3 of which are either nominalizations (C-NOUNs found in NOMLEX-PLUS: Meyers et al., 2004; Meyers, 2007) or TECH-ADJs.
3. T is a single word, a nominalization at least 11 characters long.
4. T is a multi-word sequence, ending in a common noun and containing a nominalization.

A final filter aims to distinguish named entities from in-line terms. It turns out that named entities, like jargon terms, include many out of vocabulary words. Thus we look for NEs among those PTs that remain after stage 3 and contain capitalized words (a single capital letter followed by lowercase letters). These NE filters are based on manually collected lists of named entities and nationality adjectives, as well as common NE endings. Dictionary lookup is used to assign GPE (ACE's Geopolitical Entity) to New York or American; LOC(ation) to Aegean Sea and Ural Mountains; and FAC(ility) to Panama Canal and Suez Canal. Plurals of nationality words, e.g., Americans are filtered out as non-terms. Terms are filtered by endings typically associated with non-terms, e.g., et al. signals that a potential term is actually a citation to articles. Honorifics (Esq, PhD, Jr, Snr) indicate that a phrase is probably a PER(son) NE. Finally, if at least one of the words in a multi-word term is a first or last person name, we can further filter them by the last word in the phrase. An ORGanization NE is assumed if the last word is ***agency, association, college*** or 65 other words. The words ***Heights, Township, Park,*** and others indicate GPE named entities. ***Street, Avenue,*** and ***Boulevard*** indicate LOC(ation) named entities. It turns out that 2 word capitalized structures including at least one person name are usually either ORG or GPE in our patent

TABLE 1 | State transition table for terminology chunker.

Previous POS	Current POS	Previous state	New state
Anything	POSS, other	Anything	O
O-NOUN, C-NOUN, PLUR-NOUN	ROM-NUM	B-T or I-T	E-T
Anything	PLUR-NOUN, C-NOUN, PER-NOUN, O-NOUN	B-T or I-T	I-T
Anything	ADJ, CAP-ADJ	I-T	I-T
O-Noun	CAP-ADJ, TECH-ADJ, NAT-ADJ	B-T or I-T	I-T
Anything	CAP-ADJ, TECH-ADJ, NAT-ADJ, ING-VERB, ED-VERB, C-NOUN, O-NOUN, PER-NOUN	E-T, O, Start	B-T
TECH-ADJ, NAT-ADJ, ADJ, CAP-ADJ	TECH-ADJ, NAT-ADJ, ADJ, CAP-ADJ	B-T or I-T	I-T
Everything else			O

corpus, and we maintain this ambiguity, but mark them as non-terms[9].

Identifying Terms by Abbreviations

We extract instances of abbreviations and full forms, using pattern matching similar to Schwartz and Hearst (2003) in contexts where a full form/abbreviation pair are separated by an open parentheses, e.g., *Hypertext Markup Language (HTML)*. In the simplest case, the abbreviation consists of the initials for each word of the full form (e.g., *SAS* is an abbreviation for Statistical Analysis System), but we also allow for several more complex cases. Abbreviations can skip stop words like *the, a, in, out*, *and*, others, e.g., *YHL* abbreviates *Years of Healthy Life* (no initial corresponds to the word *of*). Multiple letters can match a single word, e.g., *Hypertext* corresponds to the *HT* of *HTML*). There can be a correspondence between Greek and Roman letters, e.g., *TGF-β* abbreviates *Transforming Growth Factor Beta*). These and other special cases are all accounted for. After establishing a full-form/abbreviation correspondence, we use keyword-based heuristics and gazetteers to differentiate non-terminology abbreviation cases from terminology ones. For example, *New York University (NYU)* and *Acbel Polytech Inc. (API)*, are ruled out as terminology because the words *Inc*. and *University* indicate organizations; *British Columbia (BC)* is ruled out due to a gazetteer. Each term abbreviation (e.g., *html*) and the associated longer term (e.g., *Hypertext Markup Language*) are classified as instances of a single term (*Hypertext Markup Language*) for purposes of subsequent stages.

Summary of Stage 1

Both the terminology chunker and the abbreviation system identify terms in sentences in each document. These instances are collected and output to be used for stage 2. The chunker uses a FSM with the transitions conditioned on FPOS tags, to identify potential in-line terms. Additional filters based on linguistic features are used to identify the final in-line terms. The abbreviation system uses standard patterns to identify instances in the text where a phrase is linked to its corresponding abbreviation, both of which are likely to be either an in-line term or a NE. We use word lists and heuristics to eliminate the NE instances of abbreviations. Selecting these in-line terms is a major differentiation between our approach and other approaches. We find word sequences that are likely to be instances of terms, sequences containing nouns that are too rare to be included in a general purpose dictionary (O-Nouns) and other words that tend to be technical. Additionally, abbreviations are likely to be terms because authors tend to abbreviate important technical phrases. Together, these methods find good candidates for subsequent stages of Termolator. Arguably, this process of term candidate selection is a major differentiator between Termolator and other systems.

[9]We are currently experimenting with a modification to the system that allows the user to provide the output of a named entity tagger (or similar program) to block particular types of phrases from being considered as terms.

Other Details About Stage 1: Compound Terms and Stemming

Compound Terms

The Stage 1 system can combine instances of two adjacent or nearly adjacent inline terms to form *compound terms*. The two smaller terms are combined when they fall into one of the following 2 patterns:

1. There are 1 or 2 words between the first and second term, such that a preposition from the set {*of, for*} immediately follows the first inline term. The preposition is optionally followed by a determiner from the set {*a, the, an*}, e.g., *alignment algorithms for rna secondary structures* is a combination of the inline terms *alignment algorithms* and *rna secondary structures* (a singular form of this same term could include a determiner in the second short inline term as in *alignment algorithm for an rna secondary structure*).
2. The first and second term are one right after the other, e.g., *Post-HF event medical management* is the combination of the inline terms *Post-HF event* and *medical management*.

Both the initial in-line terms and the longer longer compound inline terms are output by the system as potential terms and are treated separately in Stage 2.

Stemming

In Stage 2, the instances of particular terms derived in stage 1 will be "counted." For purposes of counting, equivalences are established between terms that share the same lemma. Thus, we must make some assumptions about which items are regularized to the same lemma. Plural forms of terms are regularized to their singular counterparts, e.g., *Optical Character Recognition*

Systems ➔ *Optical Character Recognition System*, and thus plural and singular forms count as instances of the same term lemma. Given a noun that is also a verb, the –ing form is regularized to the singular noun, e.g., *network modeling* ➔ *network model*. Abbreviations are regularized to the fully spelled out form, e.g., *OCR* ➔ *Optical character Recognition*. Finally, compound terms with the prepositions *for* or *of* are regularized to prenominal noun modifier equivalents. Given a compound term of the form **NP1 preposition NP2**: (1) the determiner is dropped from NP2 and the final noun, if plural is converted to singular form; (2) NP2 is moved before NP1. For example, *Recognition of Optical Characters* is regularized to *Optical Character Recognition*. Thus for statistical purposes, a single lemma *Optical Character Recognition* will be correspond to instances of: *Optical Character Recognition, Optical Character Recognitions, OCR, OCRs,* and *Recognition of Optical Characters*. The output of lemmatization is included in the output of Stage 1, both as information associated with each recognized term and as a dictionary from lemmas to possible phrases that map to these lemmas. The dictionary is used to augment the final set of ranked terms (lemmas) to include the variants of each form, e.g., if *Optical Character Recognition* is in the output list, it would be associated with any variants of the term that actually occur in the input text, a subset of: {*Optical Character Recognition*, *Optical Character Recognitions*, *OCR*, *OCRs*, *Recognition of Optical Characters*}.

Applications of Stage 1 Output

As discussed in the introduction, the output of stage 1 is the input to stage 2. However, we have found other applications of inline terms, the output of stage 1. We used them as potential arguments of the Information Extraction relations discussed in Meyers et al. (2014b). Some example relations from the PubMed corpus follow:

1. found *in the* ***IκB protein****, an* ***inhibitor of NF-κB***
   - Relation: **Exemplify**, Arg_1: ***IκB protein****,* Arg_2*:* ***inhibitor of NF-κB***
   - Interpretation: Arg_1 is an instance of Arg_2
2. *a* ***necrotrophic effector system*** *that is an exciting contrast to the* ***biotrophic effector models*** *that have been intensively studied*
   - Relation: **Contrast**, Arg_1: ***necrotrophic effector system***, Arg_2: ***biotrophic effector models***
   - Interpretation: Arg_1 and Arg_2 are in contrast with each other
3. ***Bayesian networks*** *hold a considerable advantage over* ***pairwise association tests***
   - Relation**: Better than**, Arg_1: ***Bayesian networks,*** Arg_2: ***pairwise association tests***
   - Interpretation: Arg_1 is better than Arg_2 (in some respect)
4. ***housekeeping gene 36B4*** (***acidic ribosomal phosphoprotein P0***)
   - Relation: **Alias**, Arg_1: ***housekeeping gene 36B4***, Arg_2: ***acidic ribosomal phosphoprotein P0***
   - Interpretation: Arg_1 and Arg_2 are alternative names for the same concept, but neither is a shortened form (acronym or abbreviation).

Additionally, we have begun some research that uses in-line terms to improve Machine Translation (MT). It hypothesize that it is useful to treat in-line terms (and other fixed phrases like named entities) differently from other source language input. For phrase-based MT, these words are unlikely to be in the phrase table from (general domain) training data; these words are more likely than other words to be translated as themselves in the target language; these words are likely to be translated as single units (the constituent boundaries of the terms should not be interrupted by other translations) and finally, these phrases may correspond to terminology detected in the target language using terminology extraction. We are looking toward using fuzzy-match repair methods for translation of these units, along the lines of Ortega et al. (2016). More generally, inline terms appear to be good candidate entities that represent technical concepts for possibly a large variety of NLP applications.

While Stage 2 provides a way of selecting the "most important" terms for certain applications. Stage 1 provides a way of finding a large subset of terms useful for a variety of other applications, where finding only the most "important" terms is not sufficient[10].

Stage 2: Distributional Ranking

While stage 1 identifies term instances or tokens, stage 2 groups together these tokens into general types, clustering together variants of terms and representing types their common lemmas, e.g., *Optical Character Recognition* is a type that is realized in the actual texts in a variety of ways, as noted above. The term types are returned by the system in the form of a ranked list, ranking terms by how characteristic the terms are to one set of documents about a single topic (foreground), as compared to another set of documents about a diverse set of topics (background). Essentially, a highly ranked (more characteristic) term occurs much more frequently in the foreground than it does in the background. This methodology is based on many previous systems for identifying terminology (Damerau, 1993; Drouin, 2003; Navigli and Velardi, 2004; etc.) which aim to find nouns or noun sequences (N-grams or noun groups) that are the most characteristic of a topic. The output of systems of this type have been used as Information Retrieval key words (Jacquemin and Bourigault, 2003), terms to be defined in thesauri or glossaries for a particular field (Velardi et al., 2001) and terms tracked over time as part of technology forecasting (Daim et al., 2006; Babko-Malaya et al., 2015)[11].

In Stage 2, we rank our terms using a combination of three metrics: (1) a version of the standard Term Frequency Inverse Document Frequency (TFIDF) metric; (2) the Document

[10]Obtaining the inline terms is a relatively fast process, that is dominated in our implementation (timewise) by POS tagging. The later stages of Termolator are more computationally expensive.

[11]In Technology forecasting applications, systems seek to identify patterns of changing terminology usage in corpora divided by topic and by epoch. In principle, given increased usage of particular terminology over a sequence of epochs, one can predict the increasing prominence of a technology associated with that terminology.

Relevance Document Consensus (DRDC) metric (Navigli and Velardi, 2004); and (3) the Kullback-Leibler Divergence (KLD) metric (Cover and Thomas, 1991; Hisamitsu et al., 1999). The TFIDF metric selects terms specific to a domain by favoring terms that occur more frequently in the foreground (abbreviated as *Fore*) documents than they do in the background (abbreviated as *Back*).[12] The formula is:

$$TFIDF\,(t) = \frac{freqFore(t)}{freqBack(t)} * \log\left(\frac{numBackDocs}{numBackDocContains(t)}\right)$$

where freqFore(t) and freqBack(t) respectively refer to the number of times a term occurs in the foreground and background corpora. The first term is simply a ration of foreground/background frequencies. The second term is the standard inverse document frequency of a term in the background corpus (number of background documents divided by the total number of such documents containing the term). In the DRDC metric, two factors are considered: (i) document relevance (DR), which measures the specificity of a terminological candidate with respect to the foreground via comparative with the background (the same first term as in TFIDF); and (ii) document consensus (DC), which measures the distributed use of a terminological candidate in the target domain (favoring terms that occur in lots of foreground documents). The formula for DRDC is:

$$DRDC\,(t) = \frac{freqFore(t)}{freqBack(t)} * \sum_{d \in Fore} \frac{freq\,(d,t)}{freqFore\,(t)} * \log\left(\frac{freqFore(t)}{freq(t,d)}\right)$$

The KLD metric measures the difference between two probability distributions: the probability that a term will appear in the foreground corpus vs. the background corpus. The formula is[13]:

$$KLD(t) = (\log(freqFore(t)) - \log(freqBack(t))) * freqFore(t)$$

These three metrics are combined together with equal weights, ranking both the terms produced in stage 1 and substrings of those terms, producing an ordered list.

Stage 2 uses some of the same metrics as previous work, but may achieve different results due to the differences between the stage 1 output (technical noun groups or inline terms) that Termolator uses as opposed to the normal noun groups or bigrams used by previous work. In the Experiments and Evaluation section, we compare some results of running the system using different types of input terms and demonstrate that our inline terms provide better results.

Crucially, the terms that the system outputs depend on the choice of both the foreground and the background document sets. For example, a foreground of surgery patents entails that the output may include surgical terms and/or patent terms. Different backgrounds will result in different subsets of terms. Thus given, surgery patents as foreground and a general non-patent (e.g., news) corpus as background, the output would probably include some terms specific to patents in general, even if they were not related specifically to surgery. However, given a varied set of patent documents as the background, the output terms would probably mostly be about surgical matters and not include general patent terms. This corroborates with some of the experiments described in the Experiment and Results section in which we compare Termolator with Termostat, a terminology extraction system that has a distributional component similar to Termolator's, but currently uses a fixed corpus as its background corpus for all foreground corpora.

Stage 3: Well-Formedness Score and Relevance Score

The previous stages produce a ranked list of terms, the ranking derived from the distributional score, which we normalize to ***D***, a percentile score between 0 and 1. We then combine this score with other scores between 0 and 1. We multiply all the 0–1 scores together to produce a new percentile ranking. Weights can be applied as exponents on each of the scores, resulting in one aggregate score that we use for reranking the terms. However, we currently assume all weights to equal 1. We assume 2 scores, in addition to D: *W*, a well-formedness score and ***R***, a relevance score. The aggregate score which we use for reranking purposes is simply: ***D* W* R.*** Like stage 1, the stage 2 components (***W*** and ***R***) can be used separately from the other portions of Termolator, to score or rank terms entered by a user, e.g., terms produced by other terminology extraction systems.[14]

Well-Formedness Score

Our well-formedness (W) score is based on several linguistic rules and subjective evaluations about violations of those rules. Many of these linguistic rules are built into the chunking rules in stage 1 and thus the most common score for W is 1 when used as part of Termolator. However, W does contribute to the ranking and eliminates some potential terms with scores of 0 (a 0 score for D, W or R eliminates a term since these scores are combined by multiplication). We assume that applications of the following rules are reason to give a candidate term a perfect score (1.0):

- **ABBREVIATION_OR_TERM_THAT_IS_ABBREVIATED** – This rule matches terms that are either abbreviations or a full length term that has been abbreviated, e.g., *html, hypertext markup language, OCR, optical character recognition*, ...
- **Out_of_Vocabulary_Word** – This rule matches terms consisting of single words (and their plurals) that are not found in our dictionaries, e.g., *radionuclide, photoconductor*, ...
- **Hyphenated Word + OOV Noun** – This applies if a word contains one or more hyphen and the part of the word following the last hyphen would matches the conditions described in the previous bullet, e.g., *mono-axial, lens-pixel*, ...

[12] In Meyers et al. (2014a, 2015), we refer to Foreground documents as "Related Document Groups," i.e., a group of documents that are related as they are about the same topic. We also referred to some of the numbers referring to counts as total document counts, even though they actually refer to counts in the background documents.

[13] Our KLD function is a simplified version of KL Divergence.

[14] We have used these components to evaluate sets of terms that were not produced by the Termolator as part of the FUSE project. Our subjective analysis is that they can be used effectively in this way to rate or rerank such terms, but a formal evaluation is outside the scope of this paper.

These rules yield a score of **0.7:**

- **Common_Noun_Nominalization** – This means that the term is a single word, identified as a nominalization using dictionary lookup, e.g., *demagnetization, overexposure*,
- **Hyphenated Word + Nominalization** – This applies if a word contains one or more hyphen and the part of the word following the last hyphen would match the conditions described in the previous bullet, e.g., *de-escalation, cross-fertilization*

This rule gives a score of **0.3**:

- **Normal_Common_Noun_or_Number** – This means that the term consists of a single word that is either a number, a common noun, a name or a combination of numbers and letters (e.g., *ripcord, H1D2*).

The following rules have scores that vary, depending on the type of words found in the phrase:

- **Normal_NP** – This means that the term consists of a word sequence that is part of a noun group according to our chunker, described above. The score can be as high as **1.0** if the term contains an OOV words (e.g., *electrophotographic photoconductor* contains two OOV words). A noun group containing one "unnecessary" element such as a preceding adjective, would have a score of **0.5 (*acceptable organic solvent*)**. Other noun groups or noun phrases would have scores of **0.2 (*wheel drive capacity*)**.

There are several other rules which have scores of 0 associated with them including:[15]

- **Single_Word_Non_Noun** – This means that the word is identified as a non-noun, either by dictionary lookup or by simple morphological rules, e.g., we assume that an out of vocabulary word ending in *-ly* is an adverb, e.g., *downwardly, optical, tightening*
- **Bad_character** – This means that the term contains at least one character that is not either: a) a letter; b) a number; c) a space; d) a hyphen; e) a period; or f) an apostrophe, e.g., box™, *sum_l, slope* Δa
- **Contains_conjunction** – This rule matches sequences including coordinate conjunctions (*and, or, but, nor*), e.g., *or reproducing, asic or other integrated*
- **Too many verbs** – This means that the sequence contains multiple verbs, e.g., *insulating film corresponding, emitting diodes disposed*
- **Verbal or Sentential Structure** – This means that some chunking rules found a verbal constituent other than an adjective-like pre-modifier (*broken record*), e.g., *developer containing, photoelectric converting*
- **Unexpected_POS_sequence** – This applies to multi-word terms that do not fit any of the profiles above, e.g., *of the developing roll, beam area of the charged*.

[15]Some additional patterns also yield a score of 0, e.g., terms consisting of a single character.

In addition to ranking the output of Stage 1, Stage 2 also ranks highly frequent substrings of stage 1, e.g., if *intravascular balloon catheter* and *cannulated balloon catheter* are frequent terms, the system may also recognize that the common substring *balloon catheter* is a frequent term. So one function of W is to rule-out ill-formed substrings by assigning them a score of 0. For example, the noun *balloon* is a substring of *balloon catheter* (and the superstrings noted above), but is not a valid term by itself–it is just a normal, non-technical common noun. So when applied to our own stage 1 terms, $\boldsymbol{W}$ usually has a value of 1, but it assigns a score of 0 to some substrings. Intermediate values occur less frequently, but may serve to rank terms containing OOV words more highly than those well-formed terms that do not, e.g., *protective shield* has a low score (0.6) because although it is well-formed (the noun *shield* is arguably a nominalization of the verb *shield*), it does not contain any OOV words or other technical words.

Relevance Score

The relevance score is derived by searching for the term using Yahoo's search engine (powered by Microsoft Bing)[16] and applying some heuristics to the search result. This score is intended to measure the "relevance" of a term to technical literature. The Relevance Score $\boldsymbol{R} = HT^2$ where the two factors $\boldsymbol{H}$ and T are defined as follows and the weight on $\boldsymbol{T}$ was determined experimentally:

- $\boldsymbol{H}$ = the total number of hits for an exact match. The log 10 of this number (up to a maximum of 10) is normalized between 0 and 1.
- $\boldsymbol{T}$ = the percentage of the top 10 hits that are either articles or patents

The following information from a Yahoo search are used to compute this score: (1) the total number of hits; (2) a check to see if this result is based on the search or if a similar search was substituted, i.e., if the result includes the phrase ***including results for*** or the phrase ***showing results for***, then we know that our search was not matched at all and we should assume that there are 0 hits; and (3) the top 10 search results as represented by URLs, titles and summaries. If there are fewer than 10 hits, we assume that there are actually 500 hits, when calculating H. For each result, we search the URL, title and summary for key words which indicate that this hit is probably an article or a patent (*patent, article, sciencedirect, proceedings, journal, dissertation, thesis, abstract*). $\boldsymbol{T}$ is equal to the number of these search results that match, divided by 10. In practice, this heuristic seems to capture the intuition that a good term is likely to be the topic of current scientific articles or patents, i.e., that the term is relevant.

Today's web search programs (Google, Bing, etc.) find documents from a query, using a combination of standard information retrieval metrics like TF-IDF and a metric such as PageRank (Page et al., 1998) that measures how prominent

[16]In theory, a different search engine could be used instead of Yahoo. While we currently use the free version, pay versions could be substituted. In practice, some additional coding may be necessary to make the output of a new search engine compatible with Termolator.

documents are on the web. By using a web search query with our terms, we are indirectly using that search engine's prominence measure (in the current case Yahoo/Microsoft's prominence measure) and, in principle, ranking prominent terms more highly.

Runtime is a limiting factor for the Relevance scores because it takes about 0.75 s to search for each term. This means that producing Relevance scores for 30 K terms takes about 6 h, a substantial portion of the overall runtime.

EXPERIMENTS AND EVALUATION

Stage 1 Annotation and Evaluation

We evaluated Stage 1's inline terms by manually annotating all the instances of inline terms in a few documents and comparing the inline terms annotated by the human annotators with those selected by the system. For purposes of annotation, we defined an (in-line) term as a word or multi-word nominal expression that is specific to some technical sublanguage. It is conventionalized in one of the following two ways:

1. The term is defined early (possibly by being abbreviated) in the document and used repeatedly (possibly only in its abbreviated form).
2. The term is special to a particular field or subfield (not necessarily the field of the document being annotated).

It is not enough if the document contains a useful description of an object of interest– there must be some conventional, definable term that can be used and reused. Thus multi-word expressions that are defined as terms must be somewhat word-like—mere descriptions that are never reused verbatim are not terms. Justeson and Katz (1995) goes further than we do: they require that terms be reused within the document being annotated, whereas we only require that they be reused (e.g., frequent hits in a web search). Criterion 2 leaves open the question of how specific to a genre an expression must be to be considered a jargon-term. At an intuitive level, we would like to exclude words like *patient*, which occur frequently in medical texts, but are also commonly found in non-expert, everyday language. By contrast, we would like to include words like *tumor* and *chromosome*, which are more intrinsic to technical language insofar as they have specialized definitions and subtypes within medical language. To clarify, we posited that a term must be sufficiently specialized so that a typical naive adult should not be expected to know the meaning of the term. We developed 2 alternative models of a naive adult:

1. Homer Simpson, an animated TV character who caricatures the typical naive adult–the annotators invoke the question: Would Homer Simpson know what this means?
2. The Juvenile Fiction sub-corpus of the COCA: The annotators go to http://corpus.byu.edu/coca/ and search under FIC:Juvenile – a single occurrence of an expression in this corpus suggests that it is probably not a jargon-term.

In addition, several rules limited the span of terms to include the head and left modifiers that collocate with the heads. Decisions about which modifiers to include in a term were difficult. However, as this evaluation task came on the heels of the relation extraction task (Meyers et al., 2014b), we based our extent rules on the definitions and the set of problematic examples that were discussed and cataloged during that project. This essentially formed the annotation equivalent of case-law for extents.

For evaluation purposes, we annotated all the instances of inline-terms in a speech recognition patent (SRP), a sunscreen patent (SUP) and a journal article about a virus vaccine (VVA). For purposes of this task, only the longest strings need be detected, e.g., if *cannulated balloon catheter* is recognized, the substring *balloon catheter* need not be annotated separately, even though it is also a valid term. Each document was annotated by 2 people and then adjudicated by Annotator 2 after discussing controversial cases **Table 2** scores annotator 1, annotator 2 and a few versions of the system by comparing each against the answer key. The table includes number of terms in the answer key, number of matches, precision, recall and F-measure. The "strict" scores are based on exact matches between system terms and answer key terms, whereas the "sloppy" scores count as correct instances where part of a system term matches part of an answer key term (span errors). For example, given an answer key item of *cannulated balloon catheter,* the strings *balloon catheter* and *cannulated balloon* would each count as incorrect for purposes of the strict score and correct for purposes of the sloppy score.

As the SRP document was annotated first, some of specification agreement process took place after annotation and the scores for annotators are somewhat lower than for the other documents. However, Annotator 1's scores for SUP and **VVA** are good approximations of how well a human being should be expected to perform and the system's scores should be compared to Annotator 1 (i.e., accounting for the adjudicator's bias).

There are four system results: two baseline systems the results of running the system and two versions of the Stage 1 system: one admitting all potential terms (PTs) and one that filters out some of the terms with the filters described in the Stage 1 chunking section. Baseline 1 assumes terms derived by removing determiners from noun groups – we used an MEMM chunker using features from the GENIA corpus (Kim et al., 2003). That system has relatively high recall, but overgenerates, yielding a lower precision and F-measure than our full system – it is also inaccurate at determining the extent of terms. Baseline 2 restricts the noun groups from this same chunker to those with O-NOUN heads. This improves the precision at a high cost to recall. Next we ran our finite state machine to derive potential in-line terms, but we did not run the subsequent filters, and the final score is for our full system. Clearly our more complex strategy performs better than these baselines and the linguistic filters increase precision more than they reduce recall, resulting in higher F-measures (though low-precision high-recall output may be better for some applications).

Evaluation of Stages 2 and 3

We ran the complete system with 5000 patents about optical systems and components as the foreground (US patent codes 250, 349, 356, 359, 362, 385, 398, and 399) and 5,000 diverse patents as background. We collected a total of 219 K terms,

TABLE 2 | Evaluation of terminology chunking annotation and system output.

			Strict				Sloppy			
	Doc	**Terms**	**Matches**	**Prec**	**Rec**	**F**	**Terms**	**Prec**	**Rec**	**F**
Ann 1	SRP	1131	798	70.8%	70.6%	70.7%	1041	92.5%	92.0%	92.2%
	SUP	2166	1809	87.5%	83.5%	85.5%	1992	96.3%	92.0%	94.1%
	VVA	919	713	90.9%	77.6%	83.7%	762	97.2%	82.9%	89.5%
Ann 2	SRP	1131	960	98.4%	84.9%	91.1%	968	99.2%	85.6%	91.9%
	SUP	2166	1999	95.5%	92.3%	93.8%	2062	98.5%	95.2%	96.8%
	VVA	919	838	97.4%	91.2%	94.2%	855	99.4%	93.0%	96.1%
BL1	SRP	1131	602	24.3%	53.2%	33.4%	968	44.2%	96.8%	60.7%
	SUP	2166	1367	36.5%	63.1%	46.2%	1897	50.6%	87.6%	64.2%
	VVA	919	576	28.5%	62.7%	39.2%	887	44.0%	96.5%	60.4%
BL 2	SRP	1131	66	24.9%	5.8%	9.5%	151	57.0%	13.4%	21.6%
	SUP	2166	771	52.3%	35.6%	42.4%	1007	68.4%	46.5%	55.3%
	VVA	919	270	45.8%	29.4%	35.8%	392	66.5%	42.6%	51.9%
Sys W/O filter	SRP	1131	932	39.0%	82.4%	53.0%	1121	46.9%	99.1%	63.7%
	SUP	2166	1475	39.7%	68.1%	50.2%	1962	52.8%	90.6%	66.7%
	VVA	919	629	27.8%	68.4%	39.5%	900	39.8%	97.9%	56.6%
Full sys	SRP	1131	669	69.0%	59.2%	63.7%	802	82.8%	70.9%	76.4%
	SUP	2166	1193	64.7%	55.1%	59.5%	1526	82.8%	70.5%	76.1%
	VVA	919	581	62.1%	63.2%	62.7%	722	77.2%	78.6%	77.9%

TABLE 3 | System Output with aggregate scores, component scores and correctness judgements.

Rank	**Term**	**D**	**W**	**R**	**Total**	**Correct**
41	Stimulable phosphor	0.866	1	0.174	0.151	Yes
104	Ion beam profile	0.889	1	0.117	0.126	Yes
346	X-ray receiver	0.906	1	0.099	0.089	Yes
533	Wavelength-variable	0.838	1	0.091	0.076	Yes
556	Irradiation time t	0.460	1	0.163	0.075	No
1275	Quadrupole lens	0.460	1	0.113	0.052	Yes
1502	Evolution	0.439	1	0.109	0.048	No
1581	Proximity correction	0.451	1	0.103	0.046	Yes
1613	Dfb laser	0.943	1	0.049	0.046	Yes
1685	Asymmetric stress	0.493	1	0.067	0.033	Yes
3834	Panoramagram	0.483	1	0.056	0.027	Yes
4203	Crystal adjacent	0.316	1	0.080	0.025	No
4244	Single-mode optical fiber	0.875	1	0.029	0.025	Yes
4467	Total reflection plane	0.988	1	0.024	0.024	Yes
4879	Photosensitive epoxy resin	0.286	1	0.079	0.022	Yes

ranked by the stage 2 system. We selected the top 30 K of these terms and ran the stage 3 processes on these 30 K terms. We ranked these top terms 3 different ways, each time selecting a different top 5,000 terms for evaluation. We selected the top 5,000 terms after ranking these 30 K terms in the following ways: (a) according to stage 2 (Distributional Score); (b) according to the Relevance Score (c) according to the Combined Score (D*R*W). As W primarily was used to remove ill-formed examples, it was not well-suited for this test as a separate factor. For each list of 5,000 terms, we sampled 100 terms, took 20 random terms from each 20% interval, manually inspected the output, and rated each term as correct or incorrect. 71% of the terms ranked according to D only were correct; 82% of the terms ranked according to R were correct and 86% of the terms ranked according to the Combined Score were correct. While we believe that it is significant that the combined score produced the best result, it is unclear whether the fact that R alone did better than the stage 2 ranking because the R score was applied to the 30 K terms out of 219 K terms with the highest D scores. While in principle, we could run R on all 219 K terms, time constraints

make it impractical to do this, in general, for all output of our system[17].

Coverage of a term extractor is difficult to measure for terms without having a human being do the task, e.g., reading all 5,000 articles and writing out the list of terms[18]. Informally however, we have observed a significant increase in term output since we adopted the chunking model described above, compared to a previous version of the system that used a standard noun chunker. In other words, we are able to take a larger number of top ranked terms than before without a major decline in accuracy. One of the tasks for future work is to develop a good metric for measuring this.

Example Term Output From These Experiments

Table 3 provides some sample potential terms along with scores ***D, W, R*** and the aggregate score. The table is arranged in descending order by the aggregate score. These terms are excerpts from the best of the three rankings described in the previous section, i.e., the terms ordered by the total score. In the right-most column is an indication of whether or not these are valid terms, as per the judgment of one of the authors. The incorrect examples include: (a) ***irradiation time t***, which is really a variable (a particular irradiation time), not a productively used noun group that should be part of a glossary or a key word; (b) ***evolution***, a common word that is part of the general language and should no longer be relegated to a list of specialized vocabulary; and (c) ***crystal adjacent***, a word sequence that does not form a natural constituent – it is part of longer phrases like ***a one-dimensional photonic crystal adjacent to the magneto-optical metal film.*** In this sequence the word ***crystal***, is modified by a long adjectival modifier beginning with the word ***adjacent*** and it would be an error to consider this pair of words a single constituent.

Comparison With Termostat

Termostat (Drouin, 2003) is a terminology extraction tool that is readily available for public use without installation[19]. To our knowledge, Termostat is the only terminology extraction system that is both available for research purposes and that can perform essentially the same task as Termoloator[20].

- There are a number of key differences between Termolator and Termostat which may explain some of the differences in the results presented below:
- Termostat uses a single foreground document about the topic of interest. This is the only input to the system. In contrast, the Termolator uses a set of foreground documents that are about the same topic, e.g., patents that share a patent code; or other documents that are known to share subject matter
- Termostat uses one general purpose background corpus in common. This is part of the system. It does not change for different foreground corpora. In contrast, Termolator expects the user to supply a set of background documents, the documents that the foreground documents should be compared to.
- Both systems use chunking procedures to find candidate terms. The most significant difference is that Termolator's chunking procedure explicitly favors chunks containing OOV and technical words, whereas Termostat relies on standard Part of Speech tags.
- The two systems use different (but similar) distributional measures to rank terms.
- Termolator adds on additional well-formedness and relevance filters.

Termostat is easy to run. One simply uploads a file to Termostat's website and it creates a list of terms from it. For our first experiment, we attempted to simulate Termostat's use case as closely as possible. We chose a single document as the foreground: a copy of Einstein's Theory of Relativity, downloadable from Project Gutenberg[21]. We removed some initial and final meta-data from Project Gutenberg before using it. We constructed a background corpus that was as close as possible to the one used by Termostat, so Termolator would be running under similar conditions. Specifically, we used the British National Corpus for Termolator's background[22]. After running both Termolator and Termostat on these data, we manually evaluated the results, using the same technique as above. Termolator's stage 2 system generated 673 terms and stage 3 ranked the top 204 of these, since for relatively small lists of terms, the system only keeps the top 30%. Termostat output 1407 terms, of which we only ranked the top 30% or 422 terms. As before, we sampled 100 terms (20 from each fifth) and then manually rated terms as valid or invalid. We rated 53% of the Termolator and 50% of the Termostat terms as being valid terms. Given the difficulty of this annotation task, we believe that it is safe to assume that the systems had roughly the same accuracy.

[17] We evaluated the correctness of terms ourselves. We previously did some experiments in which graduate biology students evaluated our biology terms. We discontinued this practice primarily because we could not afford to have experts in all of the domains for which we had terms. In addition, the domain expertise was rarely accompanied by linguistic expertise. So the process of training domain experts to make consistent determinations about what does and does not constitute a linguistic unit was difficult. In contrast, using one set of annotators resulted in more consistent evaluation. Most unknown terms could be looked up and identified with high accuracy.

[18] There are no established sets of manually encoded data to test the system with. Note that the SemEval keyword extraction task (Kim et al., 2010) while overlapping with terminology extraction, does not capture the task we are doing here. In particular, we are not attempting to find a small number of keywords for a small number of articles, but rather large sets of terms that cover fields of study. We believe that constructing such a shared task manually would be prohibitive.

[19] http://termostat.ling.umontreal.ca/index.php?lang=en_CA

[20] Much of the work that assumes a similar terminology task either precedes Droun 2003 or is not readily available for testing purposes (Justeson and Katz, 1995; Navigli and Velardi, 2004, etc.). Other "terminology extraction" systems assume different tasks, e.g., Defminer (Jin et al., 2013) describes a task of finding terms and their term definitions from computational linguistics research papers. Kim et al. (2010) describes yet another task (key word extraction) which is similar, but not the same as the terminology extraction task described here (i.e., key words are not the same as terminology). Termostat seems to be the only currently available system that frames the terminology detection task the same way as we do.

[21] http://www.gutenberg.org/ebooks/5001.txt.utf-8

[22] The British National Corpus is described here: http://www.natcorp.ox.ac.uk/. Termostat's background corpus includes both the British National Corpus and 13.7K articles from *The Gazelle*, a Montreal newspaper. We only had access to the former, so we could not use it in the background for Termolator.

Another noticeable difference is that there were more 1-word terms in Termostat's output (31%) vs. Termolator's output (20%), especially toward the beginning of the ranking— for the first 1/5 of the terms, 45% of the Termostat terms and 10% of the Termolator terms consisted of single words. In an additional experiment, we ran the filters from Stage 3 (well-formedness and relevance) on the Termostat output and sampled 100 terms in the same manner. These terms were valid 53% of the time, the same as the run with Termolator. This suggests that if the difference in accuracies turns out to be significant, this difference may be due to the Stage 3 filters. 29% of the terms generated from this experiment were single word terms, a similar percentage as with before the application of the filter.

Next we then ran both Termolator and Termostat on some patent data. We downloaded the 2002 US patent applications from the US patent office[23]. We randomly chose a 5,000 file background corpus from these files. We also selected two sets of foreground files based on patent codes for refrigeration (062) and semiconductors (438)[24]. We selected 500 documents randomly about refrigeration and 5,000 randomly about semiconductors. We ran Termolator two times, both using the patent background corpus and once with each of the two foreground corpora. Then we endeavored to run Termostat using these two foreground corpora and Termostat's standard background corpus. Since Termostat requires a single file as input, we needed to merge these files together into two foreground files, one for each domain[25]. It was no problem to run Termostat with the Refrigeration file, but the Semiconductor file (235 mb) proved too large for the web version of Termostat. However, Patrick Drouin, the author of Termostat was kind enough to run it for us on his server. We evaluated the output files in the same manner as before. For the refrigeration topic, Termolator got 70% of the sample correct, whereas Termostat got 52% correct. For the semiconductor topic, Termolator got 79% correct and Termostat got 51% correct. For the refrigeration topic, Termolator detected 37,000 possible terms, of which 30,000 went through Stage 3 and were reranked. Then the 100 being manually scored were selected from the top 5,000 (20 randomly from the first 1,000, 20 randomly from the second 1,000, etc.). Termostat selected 11,675 possible terms, the top 30% or 3,502 were sampled for scoring (we chose the top 5,000 or the top 30%, whichever is less). For the semiconductor topic, Drouin provided us with the 3,073 terms that had at least 300 instances in the input text. We sampled the 100 terms from this group and scored them.

The first use case in which there was a single input file (*Einstein's Theory of Relativity*), Termolator and Termostat produced approximately the same quality output. However, for the second use case, involving a large set of foreground files, Termolator did noticeably better. A number of factors contributed to these differences. First of all, we have found that Termolator tends to produce a larger number of good terms than other systems[26]. We believe that our chunking system provides a larger pool of good candidates, so the distributional metrics have better input and therefore can produce a larger amount of high-quality output. Secondly, this use case fits Termolator's model better than it does Termostat's. Some of Termolator's measures test how many different files contain a term – this is not possible if the foreground and background are both single files. Thirdly, by selecting a background corpus in the patent domain, this means that many of the patent-specific terminology will be ruled out (terms about legal matters and inventions in general)[27]. In contrast, by comparing to a general purpose corpus, patent terms will naturally stand out, just as much as refrigeration or semiconductor terms. Finally, although we have shown that our Stage 3 filters improve the quality of Termolator output, we have yet to prove that they will improve the output of other systems. Our initial attempt to prove this was only suggestive, giving a probably-insignificant 3 percentage point boost to Termostat's output on the Einstein document.

Caching for Efficiency

We include caching options for several parts of Termolator that are reused when the system is run multiple times with similar types of input documents. This can substantially decrease the run time (after the first time the system is run). The following caching options have been implemented:

- ***Background Statistics***: It is common to run different foreground corpora against the same background corpus. For example, we have created foreground corpora, each based on different patent codes and thus covering different specific subject matter for those patents. We then ran these systems against a background corpus consisting of a wide variety of patents. We will choose all the patent documents from the same epoch, e.g., from the same year. It turns out that each of our distributional metrics (TFIDF, DRDC, and KLD) have some components based on the foreground and others based on the background. Specifically, for the background corpus, we only need one opportunity to count the number of times that a term occurs in the background documents and its Inverse Document Frequency or IDF (log of the number of documents containing a term divide by the number of background documents). By storing this information in a file, we can use it to calculate these metrics for terms in any new foreground file.
- **Relevance Scores:** The relevance scores for terms is another example. These scores can take as much as 0.75 s per term as they are based on web searches. However, these results will change very slowly over time. Within a fairly large time window, it is reasonable to store all relevance score calculated. Thus table look up can be used for finding relevance scores

[23] All the zip files from: https://bulkdata.uspto.gov/data/patent/application/redbook/fulltext/2002/

[24] These patent codes are part of a system used for U.S. patents until 2011. It is describe here: https://www.uspto.gov/web/patents/classification/selectnumwithtitle.htm. Starting 2011, the US switched to the world-wide CPC system.

[25] The Termolator can run on XML text, including text in the format of the U.S. patents, whereas Termostat requires plain text files. Thus in creating the input to Termostat, we combined some intermediate .txt files created by Termolator.

[26] We made some informal observations in the past when comparing results. However, until now, it has proven difficult to do a formal comparison.

[27] Additionally, some of our term filters specifically rule out known patent terms, e.g., *embodiment, claim, copyright*.

whenever possible and every newly calculated score is added to the table (and the table is stored in a file).

THE CHINESE SYSTEM

Our current Chinese Termolator implements several components parallel to the English system and we intend to implement additional components in future work. The Chinese Termolator uses an in-house CTB[28] word segmenter and part-of-speech tagger and a rule based noun group chunker, but without additional rules with regard to technical words. Stage 2 is similar to the English system in that we compare word distribution in a given domain with word distribution in a general background set and find topic words of the given domain.

One challenge for the Chinese system is that Chinese word boundaries are implicit, and are automatically induced by the word segmenter, which is prone to errors. We accordingly implemented an accessor-variety (AV) based filter (Feng et al., 2004), which calculates an accessor-variety score for each word based on the number of distinct words that appear before or after it. Character sequences with low AV scores are not independent enough, and usually should not be considered as valid Chinese words (Feng et al., 2004). We therefore filter out words whose accessor-variety scores are less than 3. We evaluated the precision of extracted terms on a set of speech processing patents: the precision was 85% for the top 20 terms and 78% for the top 50 terms. This evaluation was based on 1,100 terms extracted from 2,000 patents related to speech processing.

We developed a well-formedness-based automatic evaluation metric for Chinese terms, which follows the same spirit as the English well-formedness score. This metric penalizes noun phrases that contain non-Chinese characters, contain words that are not nouns or adjectives, contain too many single character words, or are longer than 3 characters. Since this error is exactly the sort of error that would be ruled out by the AV-based filter, we do not use it as part of our own terminology system. Rather, we use it when we are applying our filters to score term lists created externally, just as we are doing with parts of the English system.

We expect to implement a version of the Relevance Score that will work with Chinese language search engines in future work. As with the English, this will be a separable component of the system that can be applied to Chinese term lists created independently from our system.

CONCLUDING REMARKS

We have described a terminology system with state-of-the-art results for English that combines several different methods including linguistically motivated rules, a statistical distribution metric and a web-based relevance metric. We can derive at least 5,000 highly accurate (80–86%) terms from 5,000 documents about a topic. Given fewer input documents, the accuracy scores may be somewhat lower – the experiment on a single file (Einstein's Theory of Relativity) resulted in 54% accuracy and the experiment on 500 refrigeration patents resulted in 70% accuracy and the experiment with semi-conductor patetens resulted in 79% accuracy. More evaluation is necessary to determine if this is a consistent trend or is confounded by other factors, e.g., perhaps some topics are easier than others.

One important characteristic of our system is its combination of knowledge-based and statistical components. The knowledge-based components (dictionaries, manual-rule based chunkers, etc.) improve the results, but slow down the expansion of the system, e.g., the creation of systems for extracting terminology in other languages. Most alternatives involve substituting statistical components, e.g., the results of web searches for the knowledge-based components. However, Termolator already has statistical components and in future work, we would consider adding more such components. We do not see statistical and knowledge-based components to be an either-or question. Rather, we seek to combine the best knowledge-based components with the best statistical ones. For example, we have shown that a knowledge-based chunker produces better input to our distributional component than other types of input.

For future work, we are interested in improving on the one document use-case. Indeed, we imagine that it would be interesting to find the top N terms for all the single documents in a collection—the terms that represent the topic of the document. We have done some preliminary experiments with supreme court decisions and are finding this to be a challenging area.

As reported, the Chinese version of Termolator currently achieves accuracy of 78% accuracy for the first 50 terms, when run on 1100 patents. In future work, we intend to further develop the system for Chinese, possibly to include additional features similar to those currently implemented only in the English system. We are also considering, creating a version of Termolator for Spanish.

AUTHOR CONTRIBUTIONS

AM: Project lead, design, implementation, research and evaluation of all stages of English system. YH: Design, evaluation and implementation of the Chinese system. ZG: Design, implementation and research for stage 2 system. JO: Optimization and evaluation of stage 2 system. SL: Design and implementation of original Stage 2 system. AG-S: Evaluation of Stage 1 system. RG: Design and technical guidance. OB-M: Design and technical guidance, evaluation, and providing use-cases.

FUNDING

Supported, in part, by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20154.

[28]https://catalog.ldc.upenn.edu/LDC2013T21

ACKNOWLEDGMENTS

This paper combines and updates work reported in Meyers et al. (2014a, 2015). Authors of this paper hold the copyrights to these preprints. Copies of the preprints are available at: http://www.aclweb.org/anthology/W/W14/W14-6002.pdf and http://ceur-ws.org/Vol-1384/paper5.pdf.

REFERENCES

Babko-Malaya, O., Seidel, A., Hunter, D., HandUber, J., Torrelli, M., and Barlos, F. (2015). "Forecasting technology emergence from metadata and language of scientific publications and patents," in *15th International Conference on Scientometrics and Informetrics* (Istanbul).

Cover, T., and Thomas, J. A. (1991). *Elements of Information Theory*. New York, NY: Wiley-Interscience.

Daim, T. U., Rueda, G., Martin, H., and Gerdsri, P. (2006). Forecasting emerging technologies: use of bibliometrics and patent analysis. *Technol. Forecast. Soc. Change* 73, 981–1012. doi: 10.1016/j.techfore.2006.04.004

Damerau, F. J. (1993). Generating and evaluating domain-oriented multiword terms from texts. *Inform. Process. Manage.* 29, 433–447.

Drouin, P. (2003). Term extraction using non-technical corpora as a point of leverage. *Terminology* 9, 99–115. doi: 10.1075/term.9.1.06dro

Feng, H., Chen, K., Deng, X., and Zheng, W. (2004). Accessor variety criteria for chinese word extraction. *Comput. Linguist.* 30, 75–93. doi: 10.1162/089120104773633394

Hisamitsu, T., Niwa, Y., Nishioka, S., Sakurai, H., Imaichi, O., Iwayama, M., et al. (1999). "Term extraction using a new measure of term representativeness," in *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition* (Tokyo).

Jacquemin, C., and Bourigault, D. (2003). "Term extraction and automatic indexing," in *Handbook of Computational Linguistics,* ed R. Mitkov (Oxford: Oxford University Press).

Jin, Y., Kan, M., Ng, J., and He, X. (2013). "Mining scientific terms and their definitions: a study of the ACL anthology," in *EMNLP-2013* (Seattle: ACL).

Justeson, J. S., and Katz, S. M. (1995). Technical terminology: some linguistic properties and an algorithm for identification in text. *Nat. Lang. Eng.* 1, 9–27.

Kim, S. N., Medelyan, O., Kan, M. Y., and Baldwin, T. (2010). SemEval-2010 task 5: automatic keyphrase extraction from scientific articles. *SemEval* 21–26.

Kim, J.-D., Ohta, T., Tateisi, Y., Tsujii, J. (2003). GENIA corpus-a semantically annotated corpus for bio-textmining. *Bioinformatics* 19(Suppl. 1), i180–i182. doi: 10.1093/bioinformatics/btg1023

Macleod, C., Grishman, R., and Meyers, A. (1997). COMLEX Syntax. *Comp. Human.* 31, 459–481.

Macleod, C., Grishman, R., Meyers, A., Barrett, L., and Reeves, R. (1998). "Nomlex: a lexicon of nominalizations," in *Proceedings of Euralex* (Liège), 98.

Meyers, A. (2007). *Those Other NomBank Dictionaries – Manual for Dictionaries that Come with NomBank*. Available online at: http:nlp.cs.nyu.edu/meyers/nombank/nomdicts.pdf

Meyers, A., Glass, Z., Grieve-Smith, A., He, Y. S. L., and Grishman, R. (2014a). "Jargon-term extraction by chunking," in *COLING Workshop on Synchronic and Diachronic Approaches to Analyzing Technical Language* (Dublin).

Meyers, A., He, Y., Glass, Z., and Babko-Malaya, O. (2015). "The termolator: terminology recognition based on chunking, statistical and search-based scores," in *Workshop on Mining Scientific Papers: Computational Linguistics and Bibliometrics* (Istanbul).

Meyers, A., Lee, G., Grieve-Smith, A., He, Y., and Taber, H. (2014b). Annotating relations in scientific articles. *LREC*-2014.

Meyers, A., Reeves, R., Macleod, C., Szekeley, R., Zielinska, V., and Young, B. (2004). "The cross-breeding of dictionaries," in *Proceedings of LREC-2004* (Lisbon).

Navigli, R., and Velardi, P. (2004). Learning domain ontologies from document warehouses and dedicated web sites. *Comput. Linguist.* 30, 151–179. doi: 10.1162/089120104323093276

Ortega, J., Forcada, M., and Sánchez-Martinez, F. (2016). Using any translation source for fuzzy-match repair in a computer-aided translation setting. *Assoc. Mach. Trans. Am.* 1:204.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). "The pagerank citation ranking: bringing order to the web," in *Proceedings of the 7th International World Wide Web Conference* (Brisbane, QLD).

Ramshaw, L. A., and Marcus, M. P. (1995). "Text chunking using transformation-based learning," in *ACL Third Workshop on Very Large Corpora* (Cambridge, MA), 82–94.

Schwartz, A., and Hearst, M. (2003). A simple algorithm for identifying abbreviation definitions in biomedical text. *Pac. Composium Biocomput.* 451–462.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *J. Documentation* 28, 11–21.

Velardi, P., Missikoff, M., and Basili, R. (2001). "Identification of relevant terms to support the construction of domain ontologies," in *Workshop on Human Language Technology and Knowledge Management* (Toulouse).

Disclaimer: The Termolator has been approved for public release; unlimited distribution.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.