

What if we had no Wikipedia? Domain-independent Term Extraction from a Large News Corpus

Yonatan Bilu

IBM Research AI

yonatanz@il.ibm.com

Shai Gretz

IBM Research AI

avishaig@il.ibm.com

Edo Cohen

IBM Research AI

edo.cohen@ibm.com

Noam Slonim

IBM Research AI

noams@il.ibm.com

Abstract

One of the most impressive human endeavors of the past two decades is the collection and categorization of human knowledge in the free and accessible format that is Wikipedia. In this work we ask what makes a term worthy of entering this edifice of knowledge, and having a page of its own in Wikipedia? To what extent is this a natural product of on-going human discourse and discussion rather than an idiosyncratic choice of Wikipedia editors? Specifically, we aim to identify such “wiki-worthy” terms in a massive news corpus, and see if this can be done with no, or minimal, dependency on actual Wikipedia entries. We suggest a five-step pipeline for doing so, providing baseline results for all five, and the relevant datasets for benchmarking them. Our work sheds new light on the domain-specific Automatic Term Extraction problem, with the problem at hand being a domain-independent variant of it.

1 Introduction

The impact of Wikipedia on modern life in general, and NLP research in particular, can not be overstated. It is hard to imagine our world without Wikipedia - yet in this work we ask the readers to suspend their disbelief and do just that. Suppose that instead of founding Bomis, Nupedia and ultimately Wikipedia, Jimmy Wales would have stayed on as a trader in Chicago Options Associates, and only now, in 2020, would have decided to establish a free online encyclopedia. How could modern NLP algorithms and resources have assisted him in doing so? As a first step, could they automatically, and with reasonable accuracy, suggest the terms and concepts that should compose Wikipedia? Clearly, one could have constructed a close approximation of Wikipedia by piecing together information from online encyclopedias, glossaries and dictionaries. So to refine our question -

is it possible to reconstruct Wikipedia by analyzing current human discourse, and, specifically, as it is reflected in the media?

This question is closely related to Automatic Term Extraction (ATE), a line of research that can trace its roots back to [Luhn \(1957\)](#), who recognized the importance of identifying the terms in a document which will facilitate its coherent retrieval. Indeed, document-level term extraction has been one of the main tasks in Information Retrieval. However, closer to this work is domain-specific ATE, which, given a corpus of documents related to a well defined knowledge domain, aims to extract the salient terms fundamental to this domain. Work on ATE have been instrumental for facilitating downstream tasks such as indexing, mention detection ([Usbeck et al., 2015](#)), extracting textual themes ([Bawakid, 2015](#)), lexicon construction ([Velardi et al., 2008](#)), ontology learning ([Brewster et al., 2007](#)), and knowledge organization ([Chisholm et al., 2016](#)) – all within the context of a well defined domain.

Our setup is somewhat different. Given a large and diverse (newspaper) corpus, can we identify terms which represent titles of Wikipedia articles, and can we rank them by order of importance?

The automatic construction of Wikipedia entails a few sub-tasks, which we will elaborate on below. First and foremost, there is the task of **Candidate classification**. Given a term, e.g., *Artificial intelligence*, we want to predict whether it is worthy of being part of Wikipedia or not. But appearance in Wikipedia is not the only distinction we should make. For example, consider the term *State*. When one searches for *State* in Wikipedia, she may be interested in the political structure (such as the state of New York) or to the state of a computer system. These Wikipedia titles differ from regular pages; thus, we define a second task, **Ambiguity detection**, whose goal is to identify when a Wikipedia

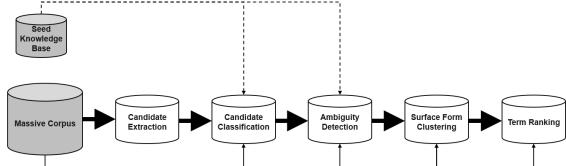


Figure 1: An overview of the suggested pipeline. A seed knowledge base is needed for the supervised classification methods described for *candidate classification* and *ambiguity detection*.

term should be a Wikipedia disambiguation page or a regular page. In addition, most Wikipedia terms do not have a single way of mentioning them in writing. For referring to *Artificial intelligence*, for example, one may use the term *Artificial Intelligence*, *AI*, or even *Cognitive systems*. Thus, we define a third task, **Surface form clustering**, whose goal is to cluster together terms which should direct to the same Wikipedia article.

For our suggested pipeline, we also include a preliminary step, **Candidate extraction**, whose goal is to extract terms from a large corpus which are likely to appear in Wikipedia, and a concluding step, **Term ranking**, for ranking terms (or clusters of terms) according to importance, to facilitate an appropriately prioritized authoring of Wikipedia pages. The full pipeline is depicted in Figure 1.

We describe baseline results for all steps, but focus on *candidate classification*, *ambiguity detection* and *surface forms clustering*, of which the latter two, due to the domain-specific nature of previous work on ATE, so far received little attention (as far as we know, *ambiguity detection* in this context is a novel task). In addition, we examine how well humans can perform the *candidate classification* step, and how the models we developed in a domain-independent manner, can lead to an effective baseline in a domain-specific context.

In subsequent sections we use the following terminology. We say that an n-gram is a *Wikipedia Term* (WT) if it corresponds to a title of a Wikipedia page, or redirects to one, and that a WT is an *Ambiguous Wikipedia Term* (AWT) if the Wikipedia page it is associated with is a disambiguation page. Conversely, we say that a WT is *Non-Ambiguous Wikipedia Term* (NAWT) if it is not an AWT. We say that two WTs are co-redirects if they are associated with the same Wikipedia page, i.e., correspond to the same Wikipedia article. In this terminology, Step 2 discerns between WTs and non-WTs; Step 3 discerns between AWTs and NAWTs; and the

clustering of Step 4 aims to cluster together two WTs iff they are co-redirects. Finally, we will also be interested in the relations between WTs and domain-specific terms, denoted here as DSTs.

2 Related Work

Interest in Automatic Term Extraction has initially been motivated by Information Retrieval needs - to identify salient terms at the document level (Luhn, 1957). By contrast, in the field of *terminology*, the focus of ATE was oriented toward the corpus as a whole. Such research started at least as early as the early 90's (Auger et al., 1991), which led to real-world solutions for this task like TERMINO (Plante and Dumas, 1998) and LEXTER (Bourigault, 1992). See (Castellví et al., 2001; Zhang et al., 2008) for a review of these earlier works and systems. As with other fields in NLP, initially such systems were mostly rule-based, with later works introducing machine-learning techniques (e.g., Conrado et al., 2013).

By and large, work on ATE focuses on the terminology of specific, well-defined domains, such as the medical domain (Fraser et al., 2019; Kim et al., 2003; Lossio-Ventura et al., 2016), natural language processing (QasemiZadeh and Schumann, 2016), and information technology services (Mophapatra et al., 2018). In the context of scientific literature ATE is sometimes used as a means toward summarizing scientific papers. For such uses, one is not interested merely in terms specific to the domain, but in those which describe specific aspects of the paper, such as terms which describe the *techniques, focus and domains* (Gupta and Manning, 2011) of the paper, and also the *applications* discussed therein (Tsai et al., 2013). This latter work also identifies the needs to cluster together different surface forms which refer to the same technique or application. They suggest a clustering algorithm based on juxtaposed citation indices, a technique specific to scientific literature. The need for clustering of surface forms is also discussed in Peng et al. (2016), where the terms of interest are mentions of events.

Domain-specific solutions, which are probably indeed of greater practical interest than the domain-independent approach, tend to face the following challenges - (a) the corpus is usually of moderate size; (b) evaluation tends to be challenging and requires manual annotation; (c) solutions tend to be domain-specific, with no single method arising

as “best practice” in the field (Zhang et al., 2016, 2018); (d) domain-specific terms need to be discerned from domain-independent terms. Hence, we hope that it might be of interest to complement domain-specific ATE with a domain-independent one.

3 Data Preparation

The data examined for candidate extraction comes from a massive English news articles corpus provided by LexisNexis.¹ This corpus contains around 400 millions articles, from which we (uniformly) sampled at random 1 million distinct sentences. During sampling we discarded sentences shorter than 10 tokens and sentences containing a newline, and kept only one copy of duplicate texts. We hereafter refer to the entire corpus as *LNC* (LexisNexis Corpus), and to the sample as *LNCS*.

Each sentence was analyzed with the SpaCy (Honnibal and Johnson, 2015) parser, and the noun phrases therein were extracted. Leading stop-words were removed from these noun phrases, and noun phrases containing only stop-words were discarded. We then kept only unigrams, bigrams, and trigrams, converting them to lowercase form.

Each n-gram was labeled for being a WT – or not – by querying English Wikipedia (online) with both the lowercase version of its surface form, and with all tokens capitalized. An n-gram identified as a WT was further marked as an AWT if the corresponding Wikipedia article contains the hallmark text “This disambiguation page lists articles associated with” (and as an NAWT, otherwise).

For unigrams and bigrams a threshold of appearing in at least 50 sentences was set, and for trigrams a threshold of 10. Table 1 lists the number of candidates so extracted, and the number of WTs and AWTs among them. Note that co-redirects are considered in this table, and during the initial 3 steps of the pipeline, as distinct WTs, with the objective of the *surface forms clustering* step being to group them together.

We denote the set of collected noun phrases as LNNP (LexisNexis Noun Phrases), and define subsets of it by $LNNP_n = \{x \in LNNP : x \text{ is a } n\text{-gram}\}$, bigram and trigrams which are frequent in LNCS, but are *not* WTs. As can be seen, all are phrases which do appear in Wikipedia. The examples depict the common reasons for this –

¹<https://www.lexisnexis.com/en-us/home.page>

Subset	# Cand.	# WT	# AWT
$LNNP_1$	6860	6306	2603
$LNNP_2$	1630	1075	120
$LNNP_3$	1315	663	20

Table 1: Number of ngrams extracted from LNCS, and the number of WTs and AWTs. Note that all AWTs are also WTs, and are included in that count.

being domain-specific (e.g., *forward-looking statements*) or being a general phrase (e.g. *tens of thousands*).

Phrase	LNCS	LNC	Wiki
fourth quarter	1234	9.3M	12164
publication name	737	4.6M	139
forward-looking statements	2476	17.8M	34
tens of thousands	244	1.7M	13208
relative price change	152	1.1M	3

Table 2: Examples of noun phrases which are common in LNCS, but are not WTs. Columns detail the number of sentences in which they appear in each of the three corpora. Note that for LNC numbers listed are in units of millions.

For the purpose of exploring *surface forms clustering* the collected data is too sparse. That is, among the collected n-grams there are very few examples of co-redirects - mostly just singular and plural forms of the same term. Moreover, this step is only relevant for WTs, and more specifically for NAWTs. Hence, for the analysis associated with this step only, we first define: $LNNP_n^*$ as the subset of $LNNP_n$ where elements are NAWTs appearing in at least 100 LNC sentences. We then augment each such set with all surface forms that are co-redirects of one of the n-grams therein, provided that they, too, appear in at least 100 LNC sentences. It should be noted that when augmenting co-redirects, there is no limit on their number of tokens. For example, co-redirects augmenting unigram candidates may (and do) contain multiple tokens. We denote such an augmentation of $LNNP_n^*$ as $LNNP_n^{aug}$.

Finally, for examining the relevance of domain-independent ATE to domain-specific ATE, we further considered the ACL RD-TEC 2.0 benchmark (QasemiZadeh and Schumann, 2016). This benchmark lists 300 abstracts of papers from the ACL anthology, manually annotated for terms consisting of specialized vocabulary related to NLP. We extracted all noun phrases from these abstracts, and all annotated terms. We define a noun phrase as a DST for this benchmark if it is identical (ignoring

case and leading stop-words) to one of the manually annotated terms.

All the above datasets will be made freely available upon publication of this work.

4 Preliminary Observations

In data preparation we followed the common wisdom of ATE research, and extracted noun phrases which are relatively common in the corpus (cf. Castellví et al., 2001). As can be seen in Table 1, this already achieves the stated goal of the *candidate extraction* step, as $LNNP$ is abundant with WTs. Hence, for the purpose of this work, we conclude that this simple method already achieves a reasonable baseline, and do not explore this step further.

Furthermore, Table 1 shows that $LNNP_1$ is composed of nearly only WTs, with a roughly even split between AWT and NAWTs. Conversely, $LNNP_{2,3}$ contain nearly no AWTs, with a roughly equally split between WTs and non-WTs. Hence, to address *candidate classification* we restrict our analysis to $LNNP_{2,3}$, while for *ambiguity detection* we restrict our analysis to $LNNP_1$.

Finally, we observe that one can also attain reasonably good results for *term ranking* based on frequency. Specifically, a commonly used metric for defining the “importance” of a Wikipedia article is the number of other articles which link to it (though many other metrics exist, e.g., Thalhammer and Rettinger (2016); Lewoniewski et al. (2016)). We find that this metric is strongly correlated with the frequency of the article’s title in LNC (Spearman rank correlation 0.77). Hence, for the purpose of obtaining baseline results for the suggested pipeline, we find this simple technique adequate, and, as with *candidate extraction*, do not explore this step further.

5 Candidates Classification

As described above, the *candidates classification* task is to determine which of the n-grams in $LNNP_{2,3}$ are WTs. We examine both supervised and unsupervised methods for this:

Relative frequency (RF): Following Nakagawa and Mori (2002), we computed for each n-gram the (log) ratio between its frequency in the corpus, and the product of the frequency of its constituent unigrams. This can be seen as a refinement of the initial *candidate selection* stage, and as an estimate for its *termhood* (cf. Kageura and Umino, 1996).

Context variance (CV): For a given n-gram, we aimed to measure the heterogeneity of the contexts in which it appears by sampling 1000 sentences containing it from LNC, and then computing the variance of the unigram-distribution over them - full details are in the appendix.

BERT: We fine-tuned BERT directly on the surface forms of the n-grams in the usual way: Data was split into train, development and evaluations sets², and the BERT model was fine tuned on the former and evaluated on the latter. Importantly, Our data sometimes contains both singular and plural forms of a noun phrase. To prevent the model from simply “copying” the label of one form in the training set to its matching form in the evaluation set, for each such singular-plural pair we discarded the less LNCS-frequent one from the analysis.

Table 3 reports the Spearman rank correlation of RF and CV measures with the ground-truth labels. In addition, we compute accuracy by considering the top k scoring n-grams as positive examples, and the remainder as negative, with k being the true number of positives. As ground truth, we consider NAWTs as positive examples, and non-WTs as negative examples. Accordingly, for CV, we consider as “top scoring” those candidates which induce lower variance, as we associate low variance with being a WT (hence the negative values in Table 3; see appendix).

Dataset	Base	RF	CV
$LNNP_2$	0.59	0.65 (0.3)	0.64 (-0.26)
$LNNP_3$	0.49	N/A	0.65 (-0.4)

Table 3: Accuracy and correlation (in parenthesis) to ground truth of unsupervised methods. *Base* is the accuracy of the majority class baseline.

Results suggest that these unsupervised methods are correlated with the ground truth with the expected sign (see appendix), but moderately so. This leads to an accuracy which is higher than the trivial baseline of predicting the majority class, especially in the case of $LNNP_3$. Interestingly, we also measured the accuracy of simply predicting the k most common candidates to be WTs. This yields an accuracy that is actually lower than the majority baseline (0.55) in the case of $LNNP_2$,

²The size of the sets was set to 20%, 20% and 60% of the data, respectively. This reflects a real-world scenario where we are given a small set of labeled terms, and aim to predict over a large set.

and essentially the same as this baseline (0.50) in the case of $LNNP_3$.

As shown in Table 4 fine tuning BERT on a small set of bigrams and trigrams (independently) yielded a much higher accuracy than the unsupervised methods. A supervised approach is not completely aligned with the question we asked at the onset of this work - how to identify WTs *de-novo*, but might be interesting in a scenario where experts identify a small number of “seed” terms as WTs, and this set is then expanded by – or with assistance of – automatic means.

Dataset	# Train	# Eval	Base	BERT
$LNNP_2$	299	870	0.63	0.82
$LNNP_3$	258	792	0.49	0.82

Table 4: Size of training and evaluation sets for fine-tuned BERT classification; accuracy of majority baseline and BERT-based classification on the latter.

With the proven success of BERT on so many NLP tasks, one might not be surprised by the attained high accuracy. Yet, keep in mind that part of BERT’s training data comes from Wikipedia. In particular, one might suspect that, as a language model, BERT assigns higher probability to bigrams and trigrams WTs, since they might be over represented in its training data relative to non WTs.

To try and control for this, we examined whether the number of Wikipedia sentences in which an n-gram appears in is a good predictor for it being a WT. We ranked the evaluation set according to this frequency, and predicted that the top k n-garms are WT and the remainder are not (with k being the number of WTs in the set). For bigrams, this yields an accuracy of 0.69, and for trigrams an accuracy of 0.79. In both cases this is higher than the majority baseline, but falls short of the BERT-based accuracy. This suggests that frequency alone can not account for this success, though further attention should be given in future work to the apparent different gap magnitude between bigrams and trigrams.

Analysis of the errors made by the fine-tuned BERT model seem to be associated with specific semantic categories. Details are in the appendix.

6 Ambiguity Detection

As described above, the *ambiguity detection* task is to determine which of the n-grams in $LNNP_1$ are AWTs, and which are NAWTs (we discard non-WTs). We applied the same unsupervised methods described in Section 5, but to no avail. The CV

method displayed low correlation with the ground truth (Spearman rank correlation 0.16), leading to an accuracy of 0.61 compared to a baseline of 0.59.

Using BERT³ lead to mixed results. In 16 of 20 runs, it failed to learn a meaningful model, and simply predicted the majority class. This led us to seek a more robust model, by leveraging the contexts in which a term appears. To do do this we employed and architecture similar to Deep Set (Zaheer et al., 2017): For each WT we extract from LNC 100 sentences in which it appears. We then process the sentences with BERT, and extract the term’s contextual vector representation . Each of these representations is a single training example used to train a fully-connected neural net. During evaluation, each of the 100 sentences extracted for a WT is classified by the model, and the predicted label for a term is determined by the average scores for these sentences (see appendix for full details).

As can be seen in Table 5, this DeepSet method, which aggregates together the different contexts in which a WT appears, did manage to surpass the majority baseline in all runs, and seems somewhat better than the more naive application of BERT. This suggests that the contexts in which a term appears, and perhaps also the relations between them, are indeed related to whether or not it is an AWT.

Method	# fail	mean acc.	max acc.
BERT	16	0.713 ± 0.015	0.729
DeepSet	0	0.725 ± 0.005	0.735

Table 5: Classification results for discerning AWTs from NAWTs (from among WTs), over 20 runs. The model is considered to have failed to learn if its accuracy is no better than the majority-class baseline (0.61). Mean accuracy and standard deviations are computed over the runs which did not fail.

7 Surface Forms Clustering

As described above, the *ambiguity detection* task is to cluster $LNNP_n^{aug}$ so that two surface forms are in the same cluster iff they are co-redirects. We consider a 2-phase approach, where the first phase is a rule-based clustering (RBC) which identifies candidates that should initially be merged. These rules have high precision, but identify only a relatively small number of candidates that should be merged. In the second phase a more standard clustering algorithm is used, alongside a more general

³Splitting $LNNP_1$ to train-dev-test with 0.2, 0.2 and 0.6 of the data, respectively.

similarity measure. This can be calibrated to yield a desired number of clusters.

Specifically, the RBC phase was done by merging candidates if they share common word-forms,⁴ ignoring case, punctuation, space, and a trailing `s` character. For example, *Artificial-intelligence*, *Artificial Intelligence* and *Artificially intelligent*, co-redirects of the Wikipedia page *Artificial intelligence*, are clustered together by this phase.

In the second phase, we examined four possible approaches for clustering candidates:

GloVe-Agg and GloVe-Har: Each candidate is represented by the average of the GloVe embeddings of its constituent unigrams (Pennington et al., 2014). After merging candidates together in the RBC step, the vector representation of this initial cluster is the average of the vectors of all merged candidates. These vectors are then clustered based on their cosine similarity. In one variant we use Agglomerative clustering⁵ to do this, and in the other Hartigan’s K-Means (Hartigan, 1975; Slonim et al., 2005).

TF-sIB: For each candidate, 100 LNC sentences in which it appears are retrieved (if several candidates were merged during RBC, one is chosen arbitrarily). The information gain of each token in these sentences w.r.t the candidate surface form is computed, and the top 2000 tokens are taken as features for a term frequency vector representation. The motivation for this approach is that co-redirects with different surface forms are likely to appear in similar contexts. These vectors are clustered using the Sequential Information Bottleneck (sIB) algorithm (Slonim et al., 2002).

BERT-Har: For each candidate, 5 LNC sentences in which it appears are retrieved (as above). For each candidate and respective sentence, we calculate the average BERT contextual token embeddings of the candidate’s constituent unigrams, from the second to last layer of BERT.⁶ We then average these vectors across the 5 sentences to obtain a single candidate representation. The rest of this approach is similar to the approach using GloVe embeddings. We use Hartigan’s K-Means for clustering, as it seemed to work somewhat better than

⁴Using the python package: https://github.com/gutfeeling/word_forms

⁵Using the scikit-learn package: <https://scikit-learn.org>

⁶In case a unigram is split to multiple word pieces, we calculate the average of its word pieces, up to 6 word pieces.

Agglomerative clustering.

In all approaches we set the number of clusters to be the number of ground-truth Wikipedia titles. In addition, we filter about 1% of the candidates for which one of the constituent unigrams is not found in GloVe’s vocabulary, or was split to word pieces by BERT’s tokenizer in a way that we were not able to merge back to a single candidate (e.g., if a candidate contained many ‘.’ symbols). The number of co-redirects and desired clusters, before and after the RBC step, is summarized in Table 6.

Dataset	# Before RBC	# After RBC	# Clusters
$LNNP_1^{aug}$	3376	2382	500
$LNNP_2^{aug}$	9387	5350	987

Table 6: Number of co-redirects for each n-gram before and after rule-based clustering, and number of clusters.

Table 7 assesses the quality of the resulting clusters using two measures: adjusted rand index (ARI) and BCubed-F1 (Amigó et al., 2009). The best results are obtained by the BERT-Har approach, by a considerable margin.

Dataset	Method	ARI	BCubed-F1
$LNNP_1^{aug}$	GloVe-Agg	0.38	0.5
	GloVe-Har	0.37	0.5
	TF-sIB	0.4	0.51
	BERT-Har	0.46	0.58
$LNNP_2^{aug}$	GloVe-Agg	0.43	0.56
	GloVe-Har	0.47	0.57
	TF-sIB	0.46	0.55
	BERT-Har	0.53	0.61

Table 7: Adjusted rand index (ARI) and BCubed-F1 of four clustering methods. Best results for co-redirects of each n-gram are in bold.

We compared a sample of generated clusters of $LNNP_2^{aug}$ for the BERT-Har and GloVe-Har methods. From this examination, the effect of the contextual representation is clear, especially for ambiguous tokens. For example, the token `common` is part of economic WTs such as `Common stock`, as well as more abstract WTs such as `Common sense`. In the output of GloVe-Har, these two WTs are found in the same cluster, as can be seen at the lower cluster of Table 8. The similarity between the two WTs is presumably a result of the shared token `common` having the same GloVe representation. However, with BERT-Har, these terms are in distinct clusters. Moreover, as can be seen at the upper cluster of the table, the candidate `Common stock` resides cor-

rectly with `Equity` shares, as they are both co-redirects to the Wikipedia title `Common stock`, even though they do not share a common token.

Where BERT-Har tends to fail is when the ground-truth resolution of Wikipedia co-redirects is too fine-grained. For example, `Common stock` and `Equity shares` are clustered together with the WT `Company stock`, which is a redirect to a different Wikipedia title, `Stock`. These subtleties are difficult to capture with the current methodology, and we leave handling them for future work. Moreover, it is not even clear whether for downstream applications, such a fine-grained distinction is desired or beneficial.

8 More on Candidate Classification

8.1 Human Performance

The answer to the question of what makes a term “Wikipedia worthy” is highly subjective, and depends on the views of Wikipedia editors. Hence, to appreciate the difficulty of the *candidate classification* task, it is interesting to see how well non-expert humans can do it.

To this end we crowd-annotated⁷ 250 of the extracted candidate bigrams for whether or not they should be a WT, with each candidate annotated by 7 annotators. The guidelines asked the annotators not to check their answer in Wikipedia, and explained that there are no wrong answers. On average, annotators achieved an accuracy of 0.76 (std=0.06). Taking the majority vote for each candidate attains an accuracy of 0.81. This suggests that individually, non-experts are better than the majority-class baseline, but not as good as the classification model, while the “wisdom of the crowd” is on par with the latter.

Initially we were concerned that although we asked annotators not to look in Wikipedia for the answers, they nonetheless will do so. Conversely, with no answers being considered wrong, and no test questions, one might be concerned that annotators would answer at random to quickly collect their pay and move on to the next task. To alleviate these concerns we published the task in a special channel, whose annotators have proven trustworthy in past tasks. Furthermore, the mediocre accuracy, and a mediocre mean inter-annotator Cohen’s Kappa of 0.47, suggests that neither of these concerns turned

out to be a major issue. On the one hand, these values are not high enough to suggest that annotators verified their work via Wikipedia, and on the other, they are high enough to suggest that at least most annotators tried to answer in earnest.

8.2 Application to Domain-Specific ATE

Can domain-independent ATE be useful for domain-specific ATE? To explore the inter-relations between the two we considered the *ACL RD-TEC 2.0* ATE benchmark (QasemiZadeh and Schumann, 2016), from which we extracted all⁸ noun phrase bigrams and deduced their label as described in Section 3. In total, 2139 bigrams were identified in this dataset, of which 1134 (53%) were implied to be DSTs. We then searched for these bigrams in Wikipedia, finding 310 of them therein, and asked whether being a WT is indicative for being a DST.

This analysis suggest that being a WT is indeed a strong indicator for being a DST: 80% of the WTs are DSTs.⁹ Yet, while the set of bigrams identified in Wikipedia has a high precision, it is relatively small, and thus inferring all bigrams outside this set to be non-DSTs yields low accuracy (0.56), due to low recall.

Can a classifier which was trained for domain-independent ATE be useful for domain-specific ATE? To test this, we used the fine-tuned BERT model described in Section 5 to predict which bigrams are DSTs, and attained an accuracy of 0.64. While this falls short of its success on identifying WTs, it does provide a clear advantage over the 0.53 accuracy of the majority-class baseline. This suggests that there may be some common linguistic characteristic to WTs and DSTs, some of which were captured by the fine-tuned BERT model. Indeed, if the model is trained on all bigrams extracted from LNCS, rather than just the ones in the train set of Section 5, the accuracy further increases to 0.71. Error analysis of these predictions appears in the appendix.

9 Discussion and Future Work

This work considers a domain-independent variant of the classical Automatic Term Extraction task.

⁸No frequency threshold was set since many terms appear only once.

⁹A cursory examination of the remaining 20% suggests that one reason for not being a DST is that the WT is an AWT, and another is that the bigram is part of a longer n-gram, which is labeled as a DST. We defer a more careful analysis to future work.

⁷Using the Figure-Eight platform - <https://www.figure-eight.com/>

Method	Cluster of WTs
BERT-Har	Preferred equity, Preferred Equity, Convertible preferred stock, Preferred stocks, Preferred stock, Convertible Preferred Stock, Preferred Stock (Preferred stock) Equity security, Capital stock, Equity securities, Company stock (Stock) Equity shares, Common stocks, Common Stock, Common stock (Common stock) Share price, Stock price, Share prices (Share price) Common equity, Common Equity (Common equity)
GloVe-Har	Common share, Common shares, Common stocks, Common stock, Common Stock (Common stock) Paine’s Common Sense, Common Sense, Thomas Paine’s Common Sense (Common Sense (pamphlet)) Common Ground, Common ground (Common Ground) Common sense, Common-sense (Common sense)

Table 8: Comparing between clusters containing the WT *Common stock*, using BERT-Har (top) and GloVe-Har (bottom). An entire cell corresponds to a single generated cluster. In parenthesis: the ground-truth Wikipedia title for the respective WTs.

This makes relevant much larger corpora than those commonly used for domain-specific ATE, and a much more comprehensive evaluation benchmark induced by Wikipedia. It also circumvents one of the main problems in ATE - that of determining *termhood* (Kageura and Umino, 1996), i.e. whether or not a term is relevant to the domain - and allows focusing on other aspects of the task. Accordingly, we address the tasks of *ambiguity detection* and *surface forms clustering*, which did not receive much attention in previous works.

As far as we know, although detecting and solving ambiguity has been the subject of much research, determining whether a WT is an AWT is in fact a novel task. Moreover, the clustering task emerging from our premise is somewhat uncommon; cluster analysis is usually applied with the goal of understanding the structure of large data by clustering it down to a manageable number of clusters. Conversely, here, clustering is a mean to an end, rather than an analysis tool. Even though the number of items to cluster is very large, the desired clustering is of numerous, small-sized clusters. As clustering algorithms and evaluation techniques have traditionally been developed in the former setting, it may be interesting to more carefully understand their applicability to this one, and perhaps learn if and how they should be adapted.

The massive corpora available for domain-independent ATE carries more potential, in terms of scale, than was realized here. We used the smaller LNCS as our starting point, and extracted a moderate number of candidates, to allow rapid explorations of the various baselines described above. However, given time and resources, one could apply the suggested pipeline to the entire LNC, as we hope to do in future work. Such an endeavor would allow not only addressing questions of accuracy,

but also of recall, that is, identifying which parts of Wikipedia are indeed reflected in LNC, and which are not. That is, while we consider our setting as domain-independent, our error analysis suggests that perhaps a better description would be multi-domain, or a mixture of domains, as future work might hopefully reveal.

Extracting a larger number of candidates would also require more careful filtering rules. When analyzing *candidate extraction*, the only processing of candidate texts was removal of stop-words. Then, in the supervised learning experiments of the two subsequent steps, further filtering was done, by keeping only one candidate from among a pair of candidates being a singular and plural form of one another. Finally, in the clustering step, and the augmentation of the data with many co-redirects, the more elaborate RBC rules were introduced. Scaling up the pipeline would require similar rules used already at the extraction stage.

In classifying WTs vs non-WTs and AWTs vs NAWTs we have relied on BERT, since it is easy to use and readily available. Hopefully, since it was trained on such a large number of examples, only part of which come from Wikipedia, that it is not too biased toward the latter. We have tried to control for that by analyzing bigram and trigram frequency in Wikipedia, and by demonstrating that the fine-tuned BERT performs well also on the unrelated benchmark of QasemiZadeh and Schumann (2016). Nonetheless, future work should do away with this potential dependency on Wikipedia.

Finally, we gave only cursory consideration to the first and last steps of the pipeline. Especially in light of our goal to expand the scale on which the pipeline operates, these steps should receive more careful attention in the future.

References

- Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.
- Pierre Auger, Patrick Drouin, et al. 1991. Automatisation des procédures de travail en terminographie. *Meta: journal des traducteurs/Meta: Translators' Journal*, 36(1):121–127.
- Abdullah Bawakid. 2015. Using wikipedia categories for discovering the themes of text documents. In *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 1, pages 452–455. IEEE.
- Didier Bourigault. 1992. Surface grammatical analysis for the extraction of terminological noun phrases. In *Proceedings of the 14th conference on Computational linguistics-Volume 3*, pages 977–981. Association for Computational Linguistics.
- Christopher Brewster, José Iria, Ziqi Zhang, Fabio Ciravegna, Louise Guthrie, and Yorick Wilks. 2007. Dynamic iterative ontology learning. *Recent Advances in Natural Language Processing (RANLP 07)*.
- M Teresa Cabré Castellví, Rosa Estopa Bagot, and Jordi Vivaldi Palatresi. 2001. Automatic term detection: A review of current systems. *Recent advances in computational terminology*, 2:53–88.
- Andrew Chisholm, Will Radford, and Ben Hachey. 2016. Discovering entity knowledge bases on the web. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 7–11.
- Merley Conrado, Thiago Pardo, and Solange Rezende. 2013. A machine learning approach to automatic term extraction using a rich feature set. In *Proceedings of the 2013 NAACL HLT Student Research Workshop*, pages 16–23.
- Kathleen C Fraser, Isar Nejadgholi, Berry De Brujin, Muqun Li, Astha LaPlante, and Khaldoun Zine El Abidine. 2019. Extracting umls concepts from medical text using general and domain-specific deep learning models. *arXiv preprint arXiv:1910.01274*.
- Sonal Gupta and Christopher D Manning. 2011. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th international joint conference on natural language processing*, pages 1–9.
- John A. Hartigan. 1975. *Clustering Algorithms*, 99th edition. John Wiley & Sons, Inc., New York, NY, USA.
- Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378.
- Kyo Kageura and Bin Umino. 1996. Methods of automatic term recognition: A review. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 3(2):259–289.
- J-D Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. 2003. Genia corpora semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.
- Włodzimierz Lewoniewski, Krzysztof Wecel, and Witold Abramowicz. 2016. Quality and importance of wikipedia articles in different languages. In *International Conference on Information and Software Technologies*, pages 613–624. Springer.
- Juan Antonio Lossio-Ventura, Clement Jonquet, Mathieu Roche, and Maguelonne Teisseire. 2016. Biomedical term extraction: overview and a new methodology. *Information Retrieval Journal*, 19(1–2):59–99.
- Hans Peter Luhn. 1957. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317.
- Prateeti Mohapatra, Yu Deng, Abhirut Gupta, Gargi Dasgupta, Amit Paradkar, Ruchi Mahindru, Daniela Rosu, Shu Tao, and Pooja Aggarwal. 2018. Domain knowledge driven key term extraction for it services. In *International Conference on Service-Oriented Computing*, pages 489–504. Springer.
- Hiroshi Nakagawa and Tatsunori Mori. 2002. A simple but powerful automatic term extraction method. In *COLING-02 on COMPUTERM 2002: second international workshop on computational terminology-Volume 14*, pages 1–7. Association for Computational Linguistics.
- Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 392–402.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Pierre Plante and Lucie Dumas. 1998. Le dépouillement terminologique assisté par ordinateur. *Terminogramme*, 46:24–28.
- Behrang QasemiZadeh and Anne-Kathrin Schumann. 2016. The acl rd-tec 2.0: A language resource for evaluating term extraction and entity recognition methods. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1862–1868.
- Noam Slonim, Gurinder Singh Atwal, Gašper Tkačik, and William Bialek. 2005. Information-based clustering. *PNAS*, 102(51):18297–18302.

Noam Slonim, Nir Friedman, and Naftali Tishby. 2002. **Unsupervised document classification using sequential information maximization.** In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 129–136, New York, NY, USA. ACM.

Andreas Thalhammer and Achim Rettinger. 2016. Pagerank on wikipedia: towards general importance scores for entities. In *European Semantic Web Conference*, pages 227–240. Springer.

Chen-Tse Tsai, Gourab Kundu, and Dan Roth. 2013. Concept-based analysis of scientific literature. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1733–1738. ACM.

Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, et al. 2015. Gerbil: general entity annotator benchmarking framework. In *Proceedings of the 24th international conference on World Wide Web*, pages 1133–1143. International World Wide Web Conferences Steering Committee.

Paola Velardi, Roberto Navigli, and D Pierluigi. 2008. Mining the web to create specialized glossaries. *IEEE Intelligent Systems*, 23(5):18–25.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. 2017. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401.

Ziqi Zhang, Jie Gao, and Fabio Ciravegna. 2016. Jate 2.0: Java automatic term extraction with apache solr. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2262–2269.

Ziqi Zhang, José Iria, Christopher Brewster, and Fabio Ciravegna. 2008. A comparative evaluation of term recognition algorithms. In *LREC*, volume 5.

Ziqi Zhang, Johann Petrak, and Diana Maynard. 2018. Adapted textrank for term extraction: a generic method of improving automatic term extraction algorithms. *Procedia Computer Science*, 137:102–108.