Automated Phrase Mining from Massive Text Corpora

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han, *Fellow, IEEE*

Abstract—As one of the fundamental tasks in text analysis, phrase mining aims at extracting quality phrases from a text corpus and has various downstream applications including information extraction/retrieval, taxonomy construction, and topic modeling. Most existing methods rely on complex, trained linguistic analyzers, and thus likely have unsatisfactory performance on text corpora of new domains and genres without extra but expensive adaption. None of the state-of-the-art models, even data-driven models, is fully automated because they require human experts for designing rules or labeling phrases. In this paper, we propose a novel framework for automated phrase mining, AutoPhrase, which supports any language as long as a general knowledge base (e.g., Wikipedia) in that language is available, while benefiting from, but not requiring, a POS tagger. Compared to the state-of-the-art methods, AutoPhrase has shown significant improvements in both effectiveness and efficiency on five real-world datasets across different domains and languages. Besides, AutoPhrase can be extended to model single-word quality phrases.

Index Terms—Automatic phrase mining, phrase mining, distant training, part-of-speech tag, multiple languages

1 Introduction

Phrase mining refers to the process of automatic extraction of high-quality phrases (e.g., scientific terms and general entity names) in a given corpus (e.g., research papers and news). Representing the text with *quality phrases* instead of *n-grams* can improve computational models for applications such as information extraction/retrieval, taxonomy construction, and topic modeling [10], [19], [21].

Almost all the state-of-the-art methods, however, require human experts at certain levels. Most existing methods [13], [31], [38] rely on *complex, trained linguistic analyzers* (e.g., dependency parsers) to locate phrase mentions, and thus may have unsatisfactory performance on text corpora of new domains and genres without extra but expensive adaption. Our latest domain-independent method SegPhrase [23] outperforms many other approaches [1], [9], [10], [13], [30], [31], [38], but still needs *domain experts* to first carefully select hundreds of varying-quality phrases from millions of candidates, and then annotate them with binary labels.

Such reliance on manual efforts by domain and linguistic experts becomes an impediment for timely analysis of massive, emerging text corpora in specific domains. An ideal *automated phrase mining* method is supposed to be *domain-independent, with minimal human effort*[1] *or reliance on linguistic analyzers*. Bearing this in mind, we propose a novel automated phrase mining framework AutoPhrase in this paper, going beyond SegPhrase, to further avoid additional manual labeling effort and enhance the performance, mainly using the following two new techniques.

1) *Robust Positive-Only Distant Training*. In fact, many high-quality phrases are freely available in general knowledge bases, and they can be easily obtained to a scale that is much larger than that produced by human experts. Domain-specific corpora usually contain some quality phrases also encoded in general knowledge bases, even when there may be no other domain-specific knowledge bases. Therefore, for distant training, we leverage the existing high-quality phrases, as available from general knowledge bases, such as Wikipedia and Freebase, to get rid of additional manual labeling effort. We independently build samples of positive labels from general knowledge bases and negative labels from the given domain corpora, and train a number of base classifiers. We then aggregate the predictions from these classifiers, whose independence helps reduce the noise from negative labels.
2) *POS-Guided Phrasal Segmentation*. There is a trade-off between the accuracy and domain-independence when incorporating linguistic processors in the phrase mining method.
   - On the domain independence side, the accuracy might be limited without linguistic knowledge.

1. The phrase "minimal human effort" indicates using only existing general knowledge bases without any other human effort.

- *J. Shang and J. Han are with the Department of Computer Science in University of Illinois at Urbana-Champaign, Champaign, IL 61820. E-mail: {shang7, hanj}@illinois.edu.*
- *M. Jiang is with the University of Notre Dame, Notre Dame, IN 46556. E-mail: mjiang2@nd.edu.*
- *X. Ren is with the University of South California, Los Angeles, CA 90089. E-mail: xiangren@usc.edu.*
- *J. Liu is with the Google Research, New York, NY. E-mail: jialu@google.com.*
- *C. R. Voss is with the US Army Research Lab, Adelph, MD 20783. E-mail: clare.r.voss.civ@mail.mil.*

Manuscript received 10 Aug. 2017; revised 15 Dec. 2017; accepted 27 Feb. 2018. Date of publication 5 Mar. 2018; date of current version 10 Sept. 2018. (Corresponding author: Jingbo Shang.)
Recommended for acceptance by Y. Chang.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2018.2812203

It is difficult to support multiple languages well, if the method is completely language-blind.
- On the accuracy side, relying on complex, trained linguistic analyzers may hurt the domain-independence of the phrase mining method. For example, it is expensive to adapt dependency parsers to special domains like clinical reports.
  As a compromise, we propose to incorporate a *pre-trained* part-of-speech (POS) tagger to further enhance the performance, when it is available for the language of the document collection. The POS-guided phrasal segmentation leverages the shallow syntactic information in POS tags to guide the phrasal segmentation model locating the boundaries of phrases more accurately.

In principle, AutoPhrase can support any language as long as a general knowledge base in that language is available. In fact, at least 58 languages have more than 100,000 articles in Wikipedia as of Feb. 2017.[2] Moreover, since pre-trained part-of-speech taggers are widely available in many languages (e.g., more than 20 languages in TreeTagger [35][3]), the POS-guided phrasal segmentation can be applied in many scenarios. It is worth mentioning that for domain-specific knowledge bases (e.g., MeSH terms in the biomedical domain) and trained POS taggers, the same paradigm applies. In this study, without loss of generality, we only assume the availability of a general knowledge base together with a pre-trained POS tagger.

As demonstrated in our experiments, AutoPhrase not only works effectively in multiple domains like scientific papers, business reviews, and Wikipedia articles, but also supports multiple languages, such as English, Spanish, and Chinese. In addition, AutoPhrase can be extended to model single-word phrases.

Our main contributions are highlighted as follows:

- We study an important problem, *automated phrase mining*, and analyze its major challenges as above.
- We propose a robust positive-only distant training method for phrase quality estimation to minimize the human effort.
- We develop a novel phrasal segmentation model to leverage POS tags to achieve further improvement, when a POS tagger is available.
- We demonstrate the robustness, accuracy, and efficiency of our method and show improvements over prior methods, with results of experiments conducted on five real-world datasets in different domains (scientific papers, business reviews, and Wikipedia articles) and different languages (English, Spanish, and Chinese).
- We successfully extend AutoPhrase to model single-word phrases, which brings about 10 to 30 percent recall improvements on different datasets.

The rest of the paper is organized as follows. Section 2 positions our work relative to existing works. Section 3 defines basic concepts including four requirements of phrases. The details of our method are covered in Section 4. Extensive experiments and case studies are presented in Section 5. Sections 6 extends AutoPhrase to model the single-word phrases and explores the effectiveness. We conclude the study in Section 7.

2 Related Work

Identifying quality phrases efficiently has become ever more central and critical for effective handling of massively increasing-size text datasets. In contrast to keyphrase extraction [17], [24], [27], [33], [36], this task goes beyond the scope of single documents and utilizes useful cross-document signals. In [4], [14], [29], interesting phrases can be queried efficiently for ad-hoc subsets of a corpus, while the phrases are based on simple frequent pattern mining methods. The natural language processing (NLP) community has conducted extensive studies typically referred to as automatic term recognition [3], [13], [31], [36], [38], for the computational task of extracting terms (such as technical phrases). This topic also attracts attention in the information retrieval (IR) community [11], [30] since selecting appropriate indexing terms is critical to the improvement of search engines where the ideal indexing units represent the main concepts in a corpus, not just literal bag-of-words.

Text indexing algorithms typically filter out stop words and restrict candidate terms to noun phrases. With predefined part-of-speech rules, one can identify noun phrases as term candidates in POS-tagged documents. Supervised noun phrase chunking techniques [6], [32], [37] exploit such tagged documents to automatically learn rules for identifying noun phrase boundaries. Other methods may utilize more sophisticated NLP technologies such as dependency parsing to further enhance the precision [18], [26]. With candidate terms collected, the next step is to leverage certain statistical measures derived from the corpus to estimate phrase quality. Some methods rely on other reference corpora for the calibration of "termhood" [38]. The dependency on these various kinds of linguistic analyzers, domain-dependent language rules, and expensive human labeling, makes it challenging to extend these approaches to emerging, big, and unrestricted corpora, which may include many different domains, topics, and languages.

To overcome this limitation, data-driven approaches opt instead to make use of frequency statistics in the corpus to address both candidate generation and quality estimation [7], [9], [10], [23], [30], [34]. They do not rely on complex linguistic feature generation, domain-specific rules or extensive labeling efforts. Instead, they rely on large corpora containing hundreds of thousands of documents to help deliver superior performance [23]. In [30], several indicators, including frequency and comparison to super/sub-sequences, were proposed to extract n-grams that reliably indicate frequent, concise concepts. Deane [9] proposed a heuristic metric over frequency distribution based on Zipfian ranks, to measure lexical association for phrase candidates. As a preprocessing step towards topical phrase extraction, El-Kishky et al. [10] proposed to mine *significant phrases* based on frequency as well as document context following a bottom-up fashion, which only considers a part of quality phrase criteria, *popularity* and *concordance*. Our previous work [23] succeeded at integrating phrase quality estimation with phrasal segmentation to further rectify the initial set of statistical features, based on local occurrence

2. https://meta.wikimedia.org/wiki/List_of_Wikipedias
3. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

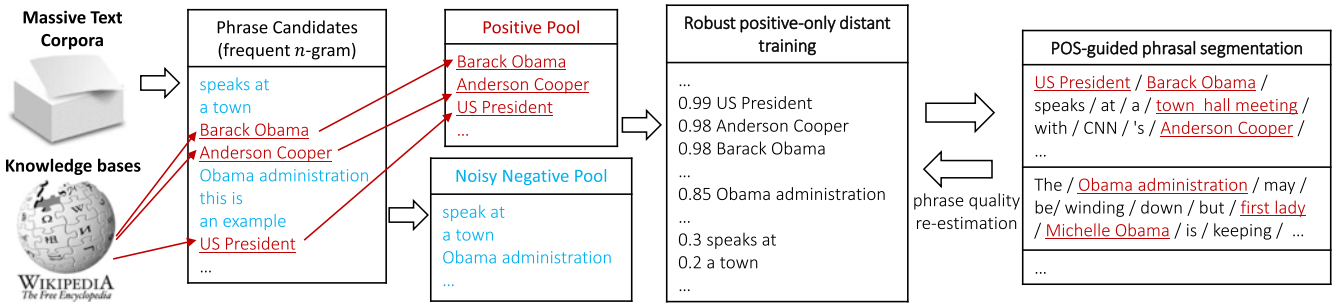


Fig. 1. The overview of AutoPhrase. The two novel techniques developed in this paper are highlighted.

context. Unlike previous methods which are purely unsupervised, [23] required a small set of phrase labels to train its phrase quality estimator. [22] follows [23] and further refines the phrasal segmentation. It is worth mentioning that all these approaches still depend on the human effort (e.g., setting domain-sensitive thresholds). Therefore, extending them to work automatically is challenging.

3 Preliminaries

The goal of this paper is to develop an automated phrase mining method to extract quality phrases from a large collection of documents without human labeling effort, and with only limited, shallow linguistic analysis. The main input to the automated phrase mining task is a corpus and a knowledge base. The input corpus is a textual word sequence in a particular language and a specific domain, of arbitrary length. The output is a ranked list of phrases with decreasing quality.

The AutoPhrase framework is shown in Fig. 1. The work flow is completely different from our previous domain-independent phrase mining method requiring human effort [23], although the phrase candidates and the features used during phrase quality (re-)estimation are the same. In this paper, we propose a robust positive-only distant training to minimize the human effort and develop a POS-guided phrasal segmentation model to improve the model performance. In this section, we briefly introduce basic concepts and components as preliminaries.

A *phrase* is defined as a sequence of words that appear consecutively in the text, forming a complete semantic unit in certain contexts of the given documents [12]. Compare to the entity, the phrase is a more general concept. Indeed, many high quality phrases are entities, like person names. However, there are also other phrases such as verb phrases. The *phrase quality* is defined to be the probability of a word sequence being a complete semantic unit, meeting the following criteria [23]:

- *Popularity*: Quality phrases should occur with sufficient frequency in the *given* document collection.
- *Concordance*: The collocation of tokens in quality phrases occurs with significantly higher probability than expected due to chance [16].
- *Informativeness*: A phrase is informative if it is indicative of a specific topic or concept.
- *Completeness*: Long frequent phrases and their subsequences within those phrases may both satisfy the 3 criteria above. A phrase is deemed complete when it can be interpreted as a complete semantic unit in some given document context. Note that a phrase and a subphrase contained within it, may both be deemed complete, depending on the context in which they appear. For example, "*relational database system*", "*relational database*" and "*database system*" can all be complete in certain context.

AutoPhrase will estimate the phrase quality based on the positive and negative pools twice, once before and once after the POS-guided phrasal segmentation. That is, the POS-guided phrasal segmentation requires an initial set of phrase quality scores; we estimate the scores based on raw frequencies beforehand; and then once the feature values have been rectified, we re-estimate the scores.

Only the phrases satisfying all above requirements are recognized as *quality phrases*.

Example 1. Examples are shown in the following table. "*strong tea*" is a quality phrase while "*heavy tea*" fails to be due to *concordance*. "*this paper*" is a *popular* and *concordant* phrase, but is not *informative* in research publication corpus. "*NP-complete in the strong sense*" is a quality phrase while "*NP-complete in the strong*" fails to be due to *completeness*.

Phrase	Quality?	Failure Criteria
strong tea	✓	N/A
heavy tea	×	concordance
this paper	×	informative
NP-complete in the strong sense	✓	N/A
NP-complete in the strong	×	completeness

To automatically mine these quality phrases, the first phase of AutoPhrase (see leftmost box in Fig. 1) establishes the set of *phrase candidates* that contains all n-grams over the minimum support threshold τ (e.g., 30) in the corpus. Here, this threshold refers to *raw frequency* of the n-grams calculated by string matching. In practice, one can also set a phrase length threshold (e.g., $n \leq 6$) to restrict the number of words in any phrase. Given a phrase candidate $w_1 w_2 \dots w_n$, its phrase quality is:

$$Q(w_1 w_2 \dots w_n) = p(\lceil w_1 w_2 \dots w_n \rfloor | w_1 w_2 \dots w_n) \in [0, 1],$$

where $\lceil w_1 w_2 \dots w_n \rfloor$ refers to the event that these words constitute a phrase. $Q(\cdot)$, also known as the *phrase quality estimator*, is initially learned from data based on statistical features,[4]

4. See https://github.com/shangjingbo1226/AutoPhrase for further details.

such as point-wise mutual information, point-wise KL divergence, and inverse document frequency, designed to model concordance and informativeness mentioned above. Note the phrase quality estimator is computed independent of POS tags. For unigrams, we simply set their phrase quality as 1.

Example 2. A good quality estimator will return $Q(\text{this paper}) \approx 0$ and $Q(\text{relational database system}) \approx 1$.

Then, to address the completeness criterion, the *phrasal segmentation* finds the best segmentation for each sentence.

Example 3. Ideal phrasal segmentation results are as follows.

#1:	*... / the / Great Firewall / is / ...*
#2:	*This / is / a / great / firewall software/ .*
#3:	*The / discriminative classifier / SVM / is / ...*

During the *phrase quality re-estimation*, related statistical features will be re-computed based on the *rectified frequency* of phrases, which means the number of times that a phrase becomes a complete semantic unit in the identified segmentation. After incorporating the rectified frequency, the phrase quality estimator $Q(\cdot)$ also models the *completeness* in addition to *concordance* and *informativeness*.

Example 4. Continuing the previous example, as shown in the following table, the *raw frequency* of the phrase

Phrase	Raw Freq	Rectified Freq
great firewall	2	1
firewall software	1	1
classifier SVM	1	0

"*great firewall*" is 2, but its *rectified frequency* is 1. Both the *raw frequency* and the *rectified frequency* of the phrase "*firewall software*" are 1. The *raw frequency* of the phrase "*classifier SVM*" is 1, but its *rectified frequency* is 0.

4 Methodology

In this section, we focus on introducing our two new techniques. First, a novel robust positive-only distant training method is developed to leverage the quality phrases in public, general knowledge bases. Second, we introduce the part-of-speech tags into the phrasal segmentation process and try to let our model take advantage of these language-dependent information, and thus perform more smoothly in different languages.

4.1 Robust Positive-Only Distant Training

To estimate the phrase quality score for each phrase candidate, our previous work [23] required domain experts to first carefully select hundreds of varying-quality phrases from millions of candidates, and then annotate them with binary labels. For example, for computer science papers, our domain experts provided hundreds of positive labels (e.g., "*spanning tree*" and "*computer science*") and negative labels (e.g., "*paper focuses*" and "*important form of*"). However, creating such a label set is expensive, especially in specialized domains like clinical reports and business reviews, because this approach provides no clues for how to identify the phrase candidates to be labeled. In this paper, we introduce a method that only utilizes existing general knowledge bases without any other human effort.

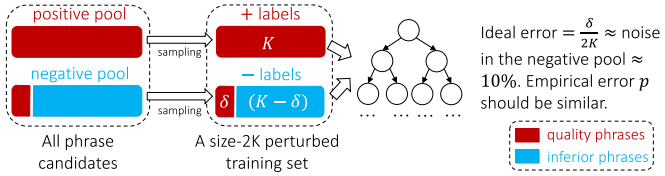


Fig. 2. The illustration of each base classifier. In each base classifier, we first randomly sample K positive and negative labels from the pools, respectively. There might be δ quality phrases among the K negative labels. An unpruned decision tree is trained based on this perturbed training set.

4.1.1 Label Pools

Public knowledge bases (e.g., Wikipedia) usually encode a considerable number of high-quality phrases in the titles, keywords, and internal links of pages. For example, by analyzing the internal links and synonyms[5] in English Wikipedia, more than a hundred thousand high-quality phrases were discovered. As a result, we place these phrases in a *positive pool*.

Knowledge bases, however, rarely, if ever, identify phrases that fail to meet our criteria, what we call *inferior phrases*. An important observation is that the number of phrase candidates, based on *n-grams* (recall leftmost box of Fig. 1), is huge and the majority of them are actually of inferior quality (e.g., "Francisco opera and"). In practice, based on our experiments, among millions of phrase candidates, usually, only about 10 percent are in good quality.[6] Therefore, phrase candidates that are derived from the given corpus but that fail to match any high-quality phrase derived from the given knowledge base, are used to populate a large but noisy *negative pool*.

4.1.2 Noise Reduction

Directly training a classifier based on the noisy label pools is not a wise choice: some phrases of high quality from the given corpus may have been missed (i.e., inaccurately binned into the negative pool) simply because they were not present in the knowledge base. Instead, we propose to utilize an ensemble classifier that averages the results of T independently trained base classifiers. As shown in Fig. 2, for each base classifier, we randomly draw K phrase candidates with replacement from the positive pool and the negative pool respectively (considering a canonical balanced classification scenario). This size-$2K$ subset of the full set of all phrase candidates is called a *perturbed training set* [5], because the labels of some (δ in the figure) quality phrases are switched from positive to negative. In order for the ensemble classifier to alleviate the effect of such noise, we need to use base classifiers with the lowest possible training errors. We grow an unpruned decision tree to the point of separating all phrases to meet this requirement. In fact, such decision tree will always reach 100 percent training accuracy when no two positive and negative phrases share identical feature representations in the perturbed training set. In

5. https://github.com/kno10/WikipediaEntities
6. This percentage is estimated when the used knowledge base is Wikipedia. It may vary when different knowledge bases are used.

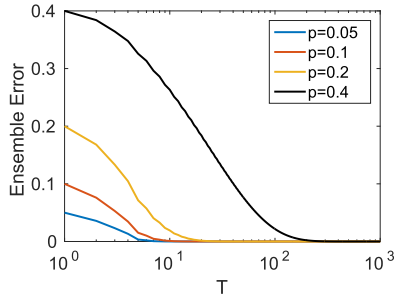


Fig. 3. Ensemble errors of different p's varying T.

this case, its ideal error is $\frac{\delta}{2K}$, which approximately equals to the proportion of switched labels among all phrase candidates (i.e., $\frac{\delta}{2K} \approx 10\%$). Therefore, the value of K is not sensitive to the accuracy of the unpruned decision tree and is fixed as 100 in our implementation. Assuming the extracted features are distinguishable between quality and inferior phrases, the empirical error evaluated on all phrase candidates, p, should be relatively small as well.

An interesting property of this sampling procedure is that the random selection of phrase candidates for building perturbed training sets creates classifiers that have statistically independent errors and similar erring probability [5], [25]. Therefore, we naturally adopt random forest [15], which is verified, in the statistics literature, to be robust and efficient. The phrase quality score of a particular phrase is computed as the proportion of all decision trees that predict that phrase is a quality phrase. Suppose there are T trees in the random forest, the ensemble error can be estimated as the probability of having more than half of the classifiers misclassifying a given phrase candidate as follows.

$$\text{ensemble\_error}(T) = \sum_{t=\lfloor 1+T/2 \rfloor}^{T} \binom{T}{t} p^t (1-p)^{T-t}.$$

From Fig 3, one can easily observe that the ensemble error is approaching 0 when T grows. In practice, T needs to be set larger due to the additional error brought by model bias. Empirical studies can be found in Fig. 8.

4.2 POS-Guided Phrasal Segmentation

Phrasal segmentation addresses the challenge of measuring *completeness* (our fourth criterion) by locating all phrase mentions in the corpus and rectifying their frequencies obtained originally via string matching.

The corpus is processed as a length-n POS-tagged word sequence $\Omega = \Omega_1 \Omega_2 \dots \Omega_n$, where Ω_i refers to a pair consisting of a word and its POS tag $\langle w_i, t_i \rangle$. A *POS-guided phrasal segmentation* is a partition of this sequence into m segments induced by a boundary index sequence $B = \{b_1, b_2, \dots, b_{m+1}\}$ satisfying $1 = b_1 < b_2 < \cdots < b_{m+1} = n+1$. The ith segment refers to $\Omega_{b_i} \Omega_{b_i+1} \dots \Omega_{b_{i+1}-1}$.

Compared to the phrasal segmentation in our previous work [23], the POS-guided phrasal segmentation addresses the completeness requirement in a *context-aware* way, instead of equivalently penalizing phrase candidates of the same length. In addition, POS tags provide shallow, language-specific knowledge, which may help boost phrase detection accuracy, especially at syntactic constituent boundaries for that language.

Given the POS tag sequence for the full length-n corpus is $t = t_1 t_2 \dots t_n$, containing the tag subsequence $t_l \dots t_{r-1}$ (denote as $t_{[l,r)}$ for clarity), the *POS quality* score for that tag subsequence is defined to be the conditional probability of its corresponding word sequence being a complete semantic unit. Formally, we have

$$T(t_{[l,r)}) = p(\lceil w_l \dots w_r \rfloor | t) \in [0, 1].$$

The POS quality score $T(\cdot)$ is designed to reward the phrases with their correctly identified POS sequences, as follows.

Example 5. Suppose the whole POS tag sequence is "NN NN NN VB DT NN". A good POS sequence quality estimator might return $T(\text{NN NN NN}) \approx 1$ and $T(\text{NN VB}) \approx 0$, where NN refers to singular or mass noun (e.g., database), VB means verb in the base form (e.g., is), and DT is for determiner (e.g., the).

The particular form of $T(\cdot)$ we have chosen is:

$$T(t_{[l,r)}) = (1 - \delta(t_{b_r-1}, t_{b_r})) \times \prod_{j=l+1}^{r-1} \delta(t_{j-1}, t_j),$$

where, $\delta(t_x, t_y)$ is the probability that the POS tag t_x is exactly precedes POS tag t_y within a phrase in the given document collection. In this formula, the first term indicates the probability that there is a phrase boundary between the words indexed $r-1$ and r, while the latter product indicates the probability that all POS tags within $t_{[l,r)}$ are in the same phrase. This POS quality score can naturally counter the bias to longer segments because $\forall i > 1$, exactly one of $\delta(t_{i-1}, t_i)$ and $(1 - \delta(t_{i-1}, t_i))$ is always multiplied no matter how the corpus is segmented. Note that the length penalty model in our previous work [23] is a special case when all values of $\delta(t_x, t_y)$ are the same.

Mathematically, $\delta(t_x, t_y)$ is defined as:

$$\delta(t_x, t_y) = p(\lceil \dots w_1 w_2 \dots \rfloor | \Omega, \text{tag}(w_1) = t_x \wedge \text{tag}(w_2) = t_y).$$

As it depends on how documents are segmented into phrases, $\delta(t_x, t_y)$ is initialized uniformly and will be learned during the phrasal segmentation.

Now, after we have both phrase quality $Q(\cdot)$ and POS quality $T(\cdot)$ ready, we are able to formally define the POS-guided phrasal segmentation model. The joint probability of a POS tagged sequence Ω and a boundary index sequence $B = \{b_1, b_2, \dots, b_{m+1}\}$ is factorized as:

$$p(\Omega, B) = \prod_{i=1}^{m} p\Big(b_{i+1}, \lceil w_{[b_i, b_{i+1})} \rfloor \Big| b_i, t\Big),$$

where $p(b_{i+1}, \lceil w_{[b_i, b_{i+1})} \rfloor | b_i, t)$ is the probability of observing a word sequence $w_{[b_i, b_{i+1})}$ as the ith quality segment given the previous boundary index b_i and the whole POS tag sequence t.

Since the phrase segments function as a constituent in the syntax of a sentence, they usually have weak dependence on each other [12], [23]. As a result, we assume these segments in the word sequence are generated one by one for the sake of both efficiency and simplicity.

For each segment, given the POS tag sequence t and the start index b_i, the generative process is defined as follows.

1) Generate the end index b_{i+1}, according to its POS quality

$$p(b_{i+1}|b_i, t) = T(t_{[b_i,b_{i+1})}).$$

2) Given the two ends b_i and b_{i+1}, generate the word sequence $w_{[b_i,b_{i+1})}$ according to a multinomial distribution over all segments of length-$(b_{i+1} - b_i)$.

$$p(w_{[b_i,b_{i+1})}|b_i, b_{i+1}) = p(w_{[b_i,b_{i+1})}|b_{i+1} - b_i).$$

3) Finally, we generate an indicator whether $w_{[b_i,b_{i+1})}$ forms a quality segment according to its quality

$$p(\lceil w_{[b_i,b_{i+1})} \rfloor | w_{[b_i,b_{i+1})}) = Q(w_{[b_i,b_{i+1})}).$$

We denote $p(w_{[b_i,b_{i+1})}|b_{i+1} - b_i)$ as $\theta_{w_{[b_i,b_{i+1})}}$ for convenience. Integrating the above three generative steps together, we have the following probabilistic factorization:

$$\begin{aligned} &p(b_{i+1}, \lceil w_{[b_i,b_{i+1})} \rfloor | b_i, t) \\ &= p(b_{i+1}|b_i, t)p(w_{[b_i,b_{i+1})}|b_i, b_{i+1})p(\lceil w_{[b_i,b_{i+1})} \rfloor | w_{[b_i,b_{i+1})}) \\ &= T(t_{[b_i,b_{i+1})})\theta_{w_{[b_i,b_{i+1})}} Q(w_{[b_i,b_{i+1})}). \end{aligned}$$

Therefore, there are three subproblems:

1) Learn θ_u for each word and phrase candidate u;
2) Learn $\delta(t_x, t_y)$ for every POS tag pair; and
3) Infer B when θ_u and $\delta(t_x, t_y)$ are fixed.

We employ the maximum a posterior principle and maximize the joint log likelihood:

$$\log p(\Omega, B) \quad = \sum_{i=1}^{m} \log p\Big(b_{i+1}, \lceil w_{[b_i,b_{i+1})} \rfloor \Big| b_t, t\Big). \tag{1}$$

Given θ_u and $\delta(t_x, t_y)$, to find the best segmentation that maximizes Equation (1), we develop an efficient dynamic programming algorithm for the POS-guided phrasal segmentation as shown in Algorithm 1.

Algorithm 1. POS-Guided Phrasal Segmentation (PGPS)

Input: Corpus $\Omega = \Omega_1\Omega_2 \dots \Omega_n$, phrase quality Q, parameters θ_u and $\delta(t_x, t_y)$.
Output: Optimal boundary index sequence B.
// $h_i \equiv \max_B \ p(\Omega_1\Omega_2 \dots \Omega_{i-1}, B|Q, \theta, \delta)$
$h_1 \leftarrow 1, h_i \leftarrow 0 \ (1 < i \leq n + 1)$
for $i = 1$ ***to*** n **do**
  for $j = i + 1$ **to** $\min(i + length\ threshold, n + 1)$ **do**
    // `Efficiently implemented via Trie.`
    if *there is no phrase starting with* $w_{[i,j)}$ **then**
      break
    // `In practice, log and addition are used to avoid underflow.`
    if $h_i \times p(j, \lceil w_{[i,j)} \rfloor | i, t_{[i,j)}) > h_j$ **then**
      $h_j \leftarrow h_i \times p(j, \lceil w_{[i,j)} \rfloor | i, t_{[i,j)})$
      $g_j \leftarrow i$
$j \leftarrow n + 1, m \leftarrow 0$
while $j > 1$ **do**
  $m \leftarrow m + 1$
  $b_m \leftarrow j$
  $j \leftarrow g_j$
return $B \leftarrow 1, b_m, b_{m-1}, \dots, b_1$

When the segmentation S and the parameter θ are fixed, the closed-form solution of $\delta(t_x, t_y)$ is:

$$\delta(t_x, t_y) = \frac{\sum_{i=1}^{m} \sum_{j=b_i}^{b_{i+1}-2} \mathbf{1}(t_j = t_x \wedge t_{j+1} = t_y)}{\sum_{i=1}^{n-1} \mathbf{1}(t_i = t_x \wedge t_{i+1} = t_y)}, \tag{2}$$

where $\mathbf{1}(\cdot)$ denotes the identity indicator. $\delta(t_x, t_y)$ is the unsegmented ratio among all $\langle t_x, t_y \rangle$ pairs in the given corpus.

Similarly, once the segmentation S and the parameter δ are fixed, the closed-form solution of θ_u can be derived as:

$$\theta_u = \frac{\sum_{i=1}^{m} \mathbf{1}(w_{[b_i,b_{i+1})} = u)}{\sum_{i=1}^{m} \mathbf{1}(b_{i+1} - b_i = |u|)}. \tag{3}$$

We can see that θ_u is the times that u becomes a complete segment normalized by the number of the length-$|u|$ segments.

As shown in Algorithm 2, we choose Viterbi Training (or Hard EM in literature [2]) to iteratively optimize parameters, because Viterbi Training converges fast and results in sparse and simple models for Hidden Markov Model-like tasks [2].

Algorithm 2. Viterbi Training (VT)

Input: Corpus Ω and phrase quality Q.
Output: θ_u and $\delta(t_x, t_y)$.
initialize θ with normalized raw frequencies
while θ_u *does not converge* **do**
  while $\delta(t_x, t_y)$ *does not converge* **do**
    $B \leftarrow$ best segmentation via Algorithm 1
    update $\delta(t_x, t_y)$ using B according to Eq. (2)
  $B \leftarrow$ best segmentation via Algorithm 1
update θ_u using B according to Eq. (3)
return θ_u and $\delta(t_x, t_y)$

4.3 Complexity Analysis

The time complexity of the most time consuming components in our framework, such as frequent n-gram, feature extraction, POS-guided phrasal segmentation, are all $O(|\Omega|)$ with the assumption that the maximum number of words in a phrase is a small constant (e.g., $n \leq 6$), where $|\Omega|$ is the total number of words in the corpus. Therefore, AutoPhrase is linear to the corpus size and thus being very efficient and scalable. Meanwhile, every component can be parallelized in an almost lock-free way grouping by either phrases or sentences.

5 Experiments

In this section, we will apply the proposed method to mine quality phrases from five massive text corpora across three domains (scientific papers, business reviews, and Wikipedia articles) and in three languages (English, Spanish, and Chinese). We compare the proposed method with many other methods to demonstrate its high performance. Then we explore the robustness of the proposed positive-only distant training and its performance against expert labeling. The advantage of incorporating POS tags in phrasal segmentation will also be proved. In the end, we present case studies.

TABLE 1
Five Real-World Massive Text Corpora in Different Domains and Multiple Languages

Dataset	Domain	Language	$\|\Omega\|$	File size	$size_p$
DBLP	Scientific Paper	English	91.6M	618 MB	29K
Yelp	Business Review	English	145.1M	749 MB	22K
EN	Wikipedia Article	English	808.0M	3.94 GB	184K
ES	Wikipedia Article	Spanish	791.2M	4.06 GB	65K
CN	Wikipedia Article	Chinese	371.9M	1.56 GB	29K

$|\Omega|$ is the total number of words. $size_p$ is the size of positive pool. To prove the domain-independence of our model, we will compare the results on the three English datasets, DBLP, Yelp, *and* EN, *as they come from different domains. To demonstrate that our model works smoothly in different languages, we will compare the results on the three Wikipedia article datasets,* EN, ES, *and* CN, *as they are of different languages.*

5.1 Datasets

To validate that the proposed positive-only distant training can effectively work in different domains and the POS-guided phrasal segmentation can support multiple languages effectively, we have five large collections of text in different domains and languages, as shown in Table 1: Abstracts of English computer science papers from *DBLP*,[7] English business reviews from *Yelp*,[8] Wikipedia articles[9] in English (*EN*), Spanish (*ES*), and Chinese (*CN*). From the existing general knowledge base Wikipedia, we extract popular mentions of entities by analyzing intra-Wiki citations within Wiki content.[10] On each dataset, the intersection between the extracted popular mentions and the generated phrase candidates forms the positive pool. Therefore, the size of positive pool may vary in different datasets of the same language.

5.2 Compared Methods

We compare *AutoPhrase* with three lines of methods as follows. Every method returns a ranked list of phrases.

SegPhrase[11]/*WrapSegPhrae*[12]: In English domain-specific text corpora, our latest work SegPhrase outperformed phrase mining [10], keyphrase extraction [30], [36], and noun phrase chunking methods. WrapSegPhrase extends SegPhrase to different languages by adding an encoding preprocessing to first transform non-English corpus using English characters and punctuation as well as a decoding postprocessing to later translate them back to the original language. Both methods require domain expert labors. For each dataset, we ask domain experts to annotate a representative set of 300 quality/interior phrases.

Parser-Based Phrase Extraction. Using complicated linguistic processors, such as parsers, we can extract minimum phrase units (e.g., NP) from the parsing trees as phrase candidates. Parsers of all three languages are available in Stanford NLP tools [8], [20], [28]. Two ranking heuristics are considered:

- *TF-IDF* ranks the extracted phrases by their term frequency and inverse document frequency in the given documents;
- *TextRank*: An unsupervised graph-based ranking model for keyword extraction [27].

Pre-trained Chinese Segmentation Models. Different from English and Spanish, phrasal segmentation in Chinese has been intensively studied because there is no space between Chinese words. The most effective and popular segmentation methods are:

- *AnsjSeg*[13] is a popular text segmentation algorithm for Chinese corpus. It ensembles statistical modeling methods of Conditional Random Fields (CRF) and Hidden Markov Models (HMMs) based on the n-gram setting;
- *JiebaPSeg*[14] is a *Chinese* text segmentation method implemented in Python. It builds a directed acyclic graph for all possible phrase combinations based on a prefix dictionary structure to achieve efficient phrase graph scanning. Then it uses dynamic programming to find the most probable combination based on the phrase frequency. For unknown phrases, an HMM-based model is used with the Viterbi algorithm.

Note that all parser-based phrase extraction and Chinese segmentation models are pre-trained based on general corpus.

To introduce a stronger baseline than SegPhrase and WrapSegPhrase, we introduce *AutoSegPhrase*, which is a hybrid of AutoPhrase and SegPhrase. *AutoSegPhrase* adopts the length penalty instead of $\delta(t_x, t_y)$, while other components are the same as AutoPhrase. Meanwhile, the comparison between AutoPhrase and AutoSegPhrase can check the effectiveness of POS-guided phrasal segmentation. In addition, AutoSegPhrase is useful when there is no POS tagger.

5.3 Experimental Settings

Implementation. The preprocessing includes tokenizers from Lucene and Stanford NLP as well as the POS tagger from TreeTagger. The pre- and post-processing are in Java, while the core functions are all implemented in C++. Experiments were all conducted on a machine with 20 cores of Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80 GHz. Our documented code package has been released and maintained in GitHub.[15]

Default Parameters. We set the minimum support threshold σ as 30. The maximum number of words in a phrase is set as 6 according to labels from domain experts. These are two parameters required by all methods. Other parameters required by compared methods were set according to the open-source tools or the original papers.

Human Annotation. We rely on human evaluators to judge the quality of the phrases which cannot be identified through any knowledge base. More specifically, on each dataset, we randomly sample 500 such phrases from the predicted phrases of each method in the experiments. These selected phrases are shuffled in a shared *pool* and evaluated by 3 reviewers independently. We allow reviewers to use search engines when unfamiliar phrases encountered. By the rule of majority voting, phrases in this pool received at least two positive annotations are *quality phrases*.

7. https://aminer.org/citation
8. https://www.yelp.com/dataset_challenge
9. https://dumps.wikimedia.org/
10. https://github.com/kno10/WikipediaEntities
11. https://github.com/shangjingbo1226/SegPhrase
12. https://github.com/remenberl/SegPhrase-MultiLingual
13. https://github.com/NLPchina/ansj_seg
14. https://github.com/fxsjy/jieba
15. https://github.com/shangjingbo1226/AutoPhrase

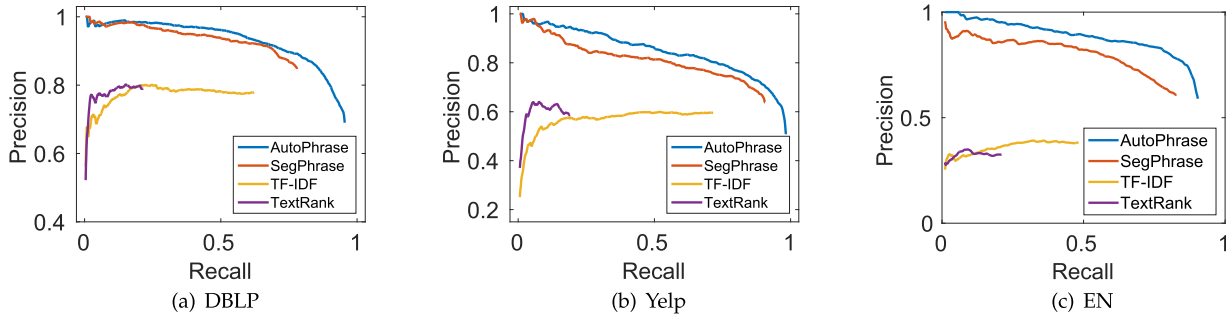


Fig. 4. Overall performance evaluation in different domains: Precision-recall curves of all methods on three English datasets of different domains evaluated by human annotation. Both AutoPhrase and SegPhrase work significantly better than other baselines. AutoPhrase always has better results than SegPhrase on English datasets, even SegPhrase is designed for English.

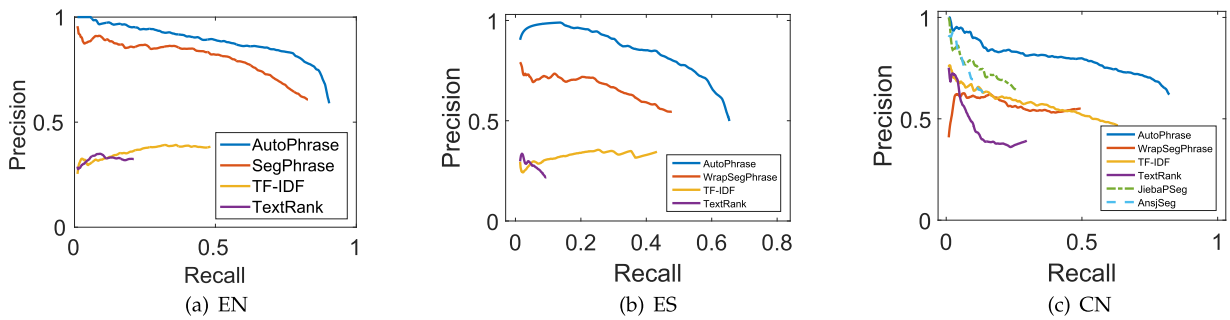


Fig. 5. Overall performance evaluation in different languages: Precision-recall curves of all methods on three Wikipedia article datasets evaluated by human annotation. The advantages of AutoPhrase over SegPhrase are more significant in non-English languages, especially on the Chinese dataset. It is worth noting that on the Chinese dataset, AutoPhrase outperforms than two popular, pre-trained Chinese phrase extraction models. This firmly demonstrates the ability of AutoPhrase to cross the language barrier.

The intra-class correlations (ICCs) are all more than 0.9 on all five datasets, which shows the agreement.

Evaluation Metrics. For a list of phrases, *precision* is defined as the number of true quality phrases divided by the number of predicted quality phrases; *recall* is defined as the number of true quality phrases divided by the total number of quality phrases. We retrieve the ranked list of the pool from the outcome of each method. When a new true quality phrase encountered, we evaluate the precision and recall of this ranked list. In the end, we plot the *precision-recall curves*. In addition, *area under the curve (AUC)* is adopted as another quantitative measure. AUC in this paper refers to the area under the precision-recall curve.

5.4 Overall Performance

Figs. 4 and 5 present the precision-recall curves of all compared methods evaluated by human annotation on five datasets. Overall, AutoPhrase performs the best, in terms of both precision and recall. Significant advantages can be observed, especially on two non-English datasets *ES* and *CN*. For example, on the *ES* dataset, the recall of AutoPhrase is about 20 percent higher than the second best method (SegPhrase) in absolute value. Meanwhile, there is a visible precision gap between AutoPhrase and the best baseline. The phrase chunking-based methods TF-IDF and TextRank work poorly, because the extraction and ranking are modeled separately and the pre-trained complex linguistic analyzers fail to extend to domain-specific corpora. TextRank usually starts with a higher precision than TF-IDF, but its recall is very low because of the sparsity of the constructed co-occurrence graph. TF-IDF achieves a reasonable recall but unsatisfactory precision. On the *CN* dataset, the pre-trained Chinese segmentation models, JiebaSeg and AnsjSeg, are very competitive, because they not only leverage training data for segmentations, but also exhaust the engineering work, including a huge dictionary for popular Chinese entity names and specific rules for certain types of entities. As a consequence, they can easily extract tons of well-known terms and people/location names. Outperforming such a strong baseline further confirms the effectiveness of AutoPhrase.

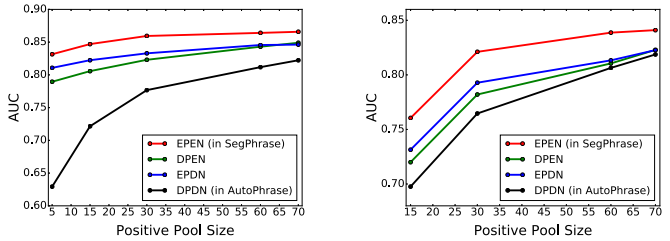
The comparison among the English datasets across three domains (i.e., scientific papers, business reviews, and Wikipedia articles) demonstrates that AutoPhrase is reasonably *domain-independent*. The performance of parser-based methods TF-IDF and TextRank depends on the rigorous degree of the documents. For example, it works well on the *DBLP* dataset but poorly on the *Yelp* dataset. However, without any human effort, AutoPhrase can work effectively on domain-specific datasets, and even outperforms SegPhrase, which is supervised by the domain experts.

The comparison among the Wikipedia article datasets in three languages (i.e., *EN*, *ES*, and *CN*) shows that, first of all, AutoPhrase *supports multiple languages*. Secondly, the advantage of AutoPhrase over SegPhrase/WrapSegPhrase is more obvious on two non-English datasets *ES* and *CN* than the *EN* dataset, which proves that *the helpfulness of introducing the POS tagger*.

As conclusions, AutoPhrase is able to support different domains and support multiple languages with minimal human effort.

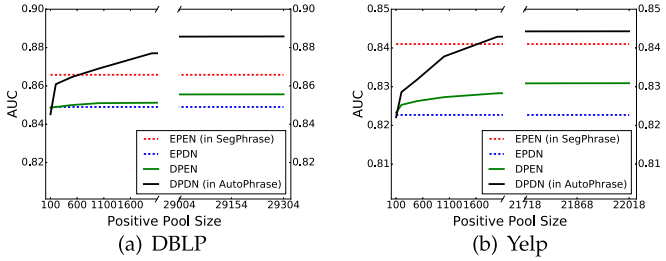
5.5 Distant Training Exploration

To compare the distant training and domain expert labeling, we experiment with the domain-specific datasets *DBLP* and



(a) DBLP (b) Yelp

Fig. 6. AUC curves of four variants *when we have **enough** positive labels in the positive pool **EP***. Overall, human annotations lead to better results because they are more clean. However, similar trends between EPEN and DPEN show that the positive pool generated from knowledge bases has reasonable quality. The similar trends between EPEN and EPDN proves that our proposed robust positive-only distant training method works well. DPDN is the worst in this case but it has a great potential to be better as the size of positive pool grows.



(a) DBLP (b) Yelp

Fig. 7. AUC curves of four variants *after we **exhaust** positive labels in the positive pool **EP***. After leveraging positive pools of enough sizes, DPDN finally becomes the best method. In the real world, the public, general knowledge bases usually have a reasonably large overlap with the domain-specific corpus, which makes DPDN more practically useful.

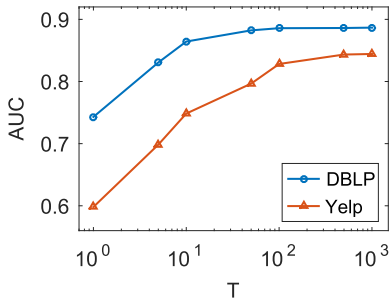


Fig. 8. AUC curves of DPDN varying T. The trends of AUC curves are similar as our theoretical analysis in Section 4.1.2. When T is large enough (e.g., 1,000), AUC scores are about 90 percent, which is very high considering the model error.

Yelp. To be fair, all the configurations in the classifiers are the same except for the label selection process. More specifically, we come up with four training pools:

1) *EP* means that domain experts give the positive pool.
2) *DP* means that a sampled subset from existing general knowledge forms the positive pool.
3) *EN* means that domain experts give the negative pool.
4) *DN* means that all *unlabeled* (i.e., not in the positive pool) phrase candidates form the negative pool.

By combining any pair of the positive and negative pools, we have four variants, *EPEN* (in SegPhrase), *DPDN* (in AutoPhrase), *EPDN*, and *DPEN*.

First of all, we evaluate the performance difference in the two positive pools. Compared to EPEN, DPEN adopts a positive pool sampled from knowledge bases instead of the well-designed positive pool given by domain experts. The negative pool *EN* is shared. As shown in Fig. 6, we vary the size of the positive pool and plot their AUC curves. We can find that EPEN outperforms DPEN and the trends of curves on both datasets are similar. Therefore, we conclude that the positive pool generated from knowledge bases has reasonable quality, although its corresponding quality estimator works slightly worse.

Secondly, we verify that whether the proposed noise reduction mechanism works properly. Compared to EPEN, EPDN adopts a negative pool of all unlabeled phrase candidates instead of the well-designed negative pool given by domain experts. The positive pool *EP* is shared. In Fig. 6, the clear gap between them and the similar trends on both datasets show that the noisy negative pool is slightly worse than the well-designed negative pool, but it still works effectively.

As illustrated in Fig. 6, DPDN has the worst performance when the size of positive pools are limited. However, distant training can generate much larger positive pools, which may significantly beyond the ability of domain experts considering the high expense of labeling. Consequently, we are curious whether the distant training can finally beat domain experts when positive pool sizes become large enough. We call the size at this tipping point as the *ideal number*.

We increase positive pool sizes and plot AUC curves of DPEN and DPDN, while EPEN and EPDN are degenerated as dashed lines due to the limited domain expert abilities. As shown in Fig. 7, with a large enough positive pool, distant training is able to beat expert labeling. On the *DBLP* dataset, the ideal number is about 700, while on the *Yelp* dataset, it becomes around 1600. Our guess is that the ideal training size is proportional to the number of words (e.g., 91.6M in *DBLP* and 145.1M in *Yelp*). We notice that compared to the corpus size, the ideal number is relatively small, which implies the distant training should be effective in many domain-specific corpora as if they overlap with Wikipedia.

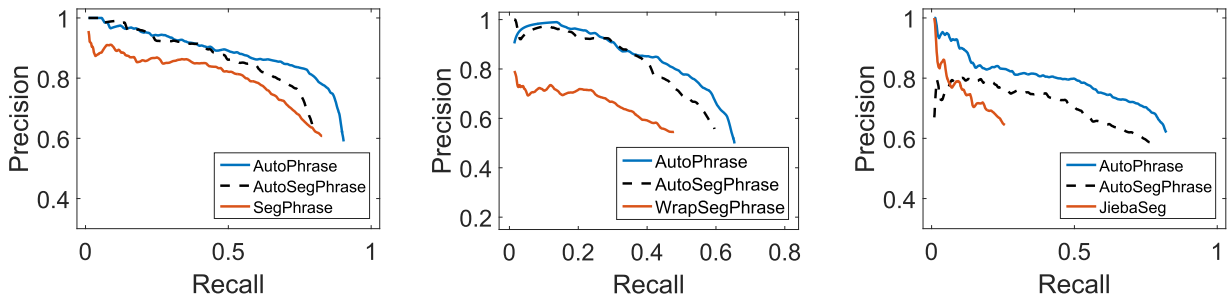
Besides, Fig. 7 shows that when the positive pool size continues growing, the AUC score increases but the slope becomes smaller. The performance of distant training will be finally stable when a relatively large number of quality phrases were fed.

We are curious how many trees (i.e., T) is enough for DPDN. We increase T and plot AUC curves of DPDN. As shown in Fig. 8, on both datasets, as T grows, the AUC scores first increase rapidly and later the speed slows down gradually, which is consistent with the theoretical analysis in Section 4.1.2.

5.6 POS-Guided Phrasal Segmentation

We are also interested in how much performance gain we can obtain from incorporating POS tags in this segmentation model, especially for different languages. We select Wikipedia article datasets in three different languages: *EN*, *ES*, and *CN*.

Fig. 9 compares the results of AutoPhrase and AutoSegPhrase, with the best baseline methods as references. AutoPhrase outperforms AutoSegPhrase even on the English dataset *EN*, though it has been shown the length penalty works reasonably well in English [23]. The Spanish dataset *ES* has similar observation. Moreover, the advantage of



(a) EN (b) ES (c) CN

Fig. 9. Comparison between phrase mining methods with/without POS tags (AutoPhrase and AutoSegPhrase) as input. Datasets in different languages are used for the comparison. The best baseline in each dataset is provided as a reference. The results show POS-guided phrasal segmentation works more smoothly in different languages. As the original segmentation method without POS information is designed for English, it works well for English and Spanish but relatively poorly on the Chinese data.

TABLE 2
The Results of AutoPhrase on the *EN* and *CN* datasets, with Translations and Explanations for Chinese Phrases

	EN	*CN*	
Rank	Phrase	Phrase	Translation (Explanation)
1	Elf Aquitaine	江苏 舜 天	(the name of a soccer team)
2	Arnold Sommerfeld	苦 艾 酒	Absinthe
3	Eugene Wigner	白发 魔 女	(the name of a novel/TV-series)
4	Tarpon Springs	笔记 型 电脑	notebook computer, laptop
5	Sean Astin	党委 书记	Secretary of Party Committee
...	...	...	...
20,001	ECAC Hockey	非洲 国家	African countries
20,002	Sacramento Bee	左翼 党	The Left (German: Die Linke)
20,003	Bering Strait	菲 沙 河谷	Fraser Valley
20,004	Jacknife Lee	海马 体	Hippocampus
20,005	WXYZ-TV	斋 贺光希	Mitsuki Saiga (a voice actress)
...	...	...	...
99,994	John Gregson	计算机 科学技术	Computer Science and Technology
99,995	white-tailed eagle	恒 天然	Fonterra (a company)
99,996	rhombic dodecahedron	中国 作家 协会 副 主席	The Vice President of Writers Association of China
99,997	great spotted woodpecker	维他命 b	Vitamin B
99,998	David Manners	舆论 导向	controlled guidance of the media
...	...	...	...

The whitespaces on the CN *dataset are inserted by the Chinese tokenizer. It worths a mention that the general knowledge base only provides about 29K quality phrases in the positive pool and* AutoPhrase *is able to discover new quality phrases even in the rank of 100K. This implies that* AutoPhrase *has a power to discover more than 200 percent new quality phrases than the provided positive pool.*

AutoPhrase becomes more significant on the *CN* dataset, indicating the poor generality of length penalty.

In summary, thanks to the extra context information and syntactic information for the particular language, incorporating POS tags during the phrasal segmentation can work better than equally penalizing phrases of the same length.

5.7 Case Study

We present a case study about the extracted phrases as shown in Table 2. The top ranked phrases are mostly named entities, which makes sense for the Wikipedia article datasets. Even in the long tail part, there are still many high-quality phrases. For example, we have the ⌈great spotted woodpecker⌋ (a type of birds) and ⌈计算机 科学技术⌋ (i.e., Computer Science and Technology) ranked about 100,000. In fact, we have more than 345 K and 116 K phrases with a phrase quality higher than 0.5 on the *EN* and *CN* datasets respectively.

5.8 Efficiency Evaluation

To study both time and memory efficiency, we choose the three largest datasets: *EN*, *ES*, and *CN*.

Fig. 10a and 10b evaluate the running time and the peak memory usage of AutoPhrase using 10 threads on different proportions of three datasets respectively. Both time and memory are linear to the size of text corpora. Moreover, AutoPhrase can also be parallelized in an almost lock-free way and shows a linear speedup in Fig. 10c.

Besides, compared to the previous state-of-the-art phrase mining method SegPhrase and its variants WrapSegPhrase on three datasets, as shown in Table 3, AutoPhrase achieves about 8 to 11 times speedup and about 5 to 7 times memory usage improvement. These improvements are made by a more efficient indexing and a more thorough parallelization.

6 Single-Word Phrase Extension

AutoPhrase can be extended to model single-word phrases, which can gain about 10 to 30 percent recall improvements

TABLE 3
Efficiency Comparison between AutoPhrase and SegPhrase/WrapSegPhrase Utilizing 10 Threads

	EN		ES		CN	
	Time (mins)	Memory (GB)	Time (mins)	Memory (GB)	Time (mins)	Memory (GB)
AutoPhrase	32.77	13.77	54.05	16.42	9.43	5.74
(Wrap)SegPhrase	369.53	97.72	452.85	92.47	108.58	35.38
Speedup/Saving	11.27	86%	8.37	82%	11.50	83%

The difference is mainly caused by a more efficient indexing and a more thorough parallelization.

on different datasets. To study the effect of modeling quality single-word phrases, we choose the three Wikipedia article datasets in different languages: *EN*, *ES*, and *CN*.

6.1 Quality Estimation

In *the paper*, the definition of quality phrases and the evaluation only focus on multi-word phrases. In linguistic analysis, however, a phrase is not only a group of multiple words, but also possibly a single word, as long as it functions as a constituent in the syntax of a sentence [12]. As a great portion (ranging from 10 to 30 percent on different datasets based on our experiments) of high-quality phrases, we should take single-word phrases (e.g., ⌈UIUC⌋, ⌈Illinois⌋, and ⌈USA⌋) into consideration as well as multi-word phrases to achieve a high recall in phrase mining.

Considering the criteria of quality phrases, because single-word phrases cannot be decomposed into two or more parts, the *concordance* and *completeness* are no longer definable. Therefore, we revise the requirements for *quality single-word phrases* as below.

- *Popularity*: Quality phrases should occur with sufficient frequency in the given document collection.
- *Informativeness*: A phrase is informative if it is indicative of a specific topic or concept.
- *Independence*: A quality single-word phrase is more likely a complete semantic unit in the given documents.

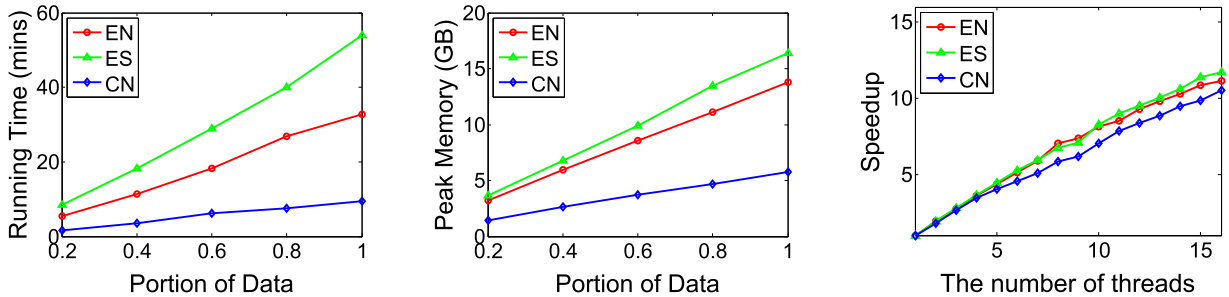
Only single-word phrases satisfying all *popularity*, *independence*, and *informativeness* requirements are recognized as quality single-word phrases.

Example 6. Examples are shown in the following table. "*UIUC*" is a quality single-word phrase. "*this*" is

Single-Word Phrase	Quality?	Failure Reason
UIUC	✓	N/A
this	×	informativeness
united	×	independence

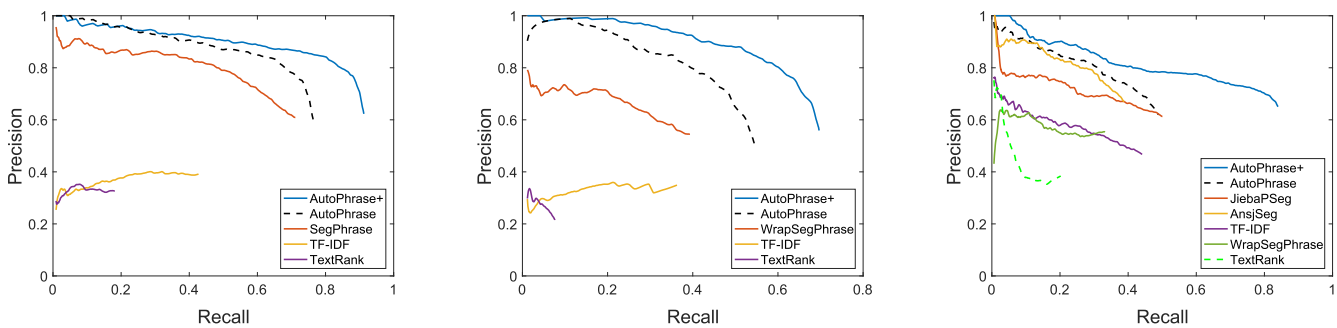
not a quality phrase due to its low informativeness. "*united*", usually occurring within other quality multi-word phrases such as "*United States*", "*United Kingdom*", "*United Airlines*", and "*United Parcel Service*", is not a quality single-word phrase, because its independence is not enough.

After the phrasal segmentation, in replacement of concordance features, the *independence feature* is added for single-word phrases. Formally, it is the ratio of the rectified



(a) Running Time (b) Peak Memory (c) Multi-threading

Fig. 10. Efficiency evaluation of AutoPhrase on the three largest datasets. Both the running time and the peak memory are linear to the corpus size. Because of an almost lock-free parallelized implementation, the multi-threading speedup is close to linear.



(a) EN (b) ES (c) CN

Fig. 11. Precision-recall curves evaluated by human annotation with both single-word and multi-word phrases in pools. The most significant recall gap can be observed in the Chinese dataset because the ratio of quality single-word phrases is highest in Chinese.

frequency of a single-word phrase given the phrasal segmentation over its raw frequency. Quality single-word phrases are expected to have large values. For example, "*united*" is likely to an almost zero ratio.

We use *AutoPhrase+* to denote the extended AutoPhrase with quality single-word phrase estimation.

6.2 Experiments

We have a similar human annotation as that in *the paper*. Differently, we randomly sampled 500 Wiki-uncovered phrases from the returned phrases (*both single-word and multi-word phrases*) of each method in experiments of *the paper*. Therefore, we have *new pools* on the *EN*, *ES*, and *CN* datasets. The intra-class correlations are all more than 0.9, which shows the agreement.

Fig. 11 compare all methods based these new pools. Note that all methods except for SegPhrase/WrapSegPhrase extract single-word phrases as well.

Significant recall advantages can be always observed on all *EN*, *ES*, and *CN* datasets. The recall differences between AutoPhrase+ and AutoPhrase, ranging from 10 to 30 percent sheds light on the importance of modeling single-word phrases. Across two Latin language datasets, *EN* and *ES*, AutoPhrase+ and AutoPhrase overlaps in the beginning, but later, the precision of AutoPhrase drops earlier and has a lower recall due to the lack of single-word phrases. On the *CN* dataset, AutoPhrase+ and AutoPhrase has a clear gap even in the very beginning, which is different from the trends on the *EN* and *ES* datasets, which reflects that single-word phrases are more important in Chinese. The major reason behind is that there are a considerable number of high-quality phrases (e.g., person names) in Chinese have only one token after tokenization.

7 Conclusions

In this paper, we present an automated phrase mining framework with two novel techniques: the robust positive-only distant training and the POS-guided phrasal segmentation incorporating part-of-speech tags, for the development of an *automated phrase mining* framework AutoPhrase. Our extensive experiments show that AutoPhrase is domain-independent, outperforms other phrase mining methods, and supports multiple languages (e.g., English, Spanish, and Chinese) effectively, with minimal human effort.

Besides, the inclusion of quality single-word phrases (e.g., ⌈UIUC⌋ and ⌈USA⌋) which leads to about 10 to 30 percent increased recall and the exploration of better indexing strategies and more thorough parallelization, which leads to about 8 to 11 times running time speedup and about 80 to 86 percent memory usage saving over SegPhrase. Interested readers may try our released code at GitHub.

For future work, it is interesting to (1) refine quality phrases to entity mentions, (2) apply AutoPhrase to more languages, such as Japanese, and (3) for those languages without general knowledge bases, seek an unsupervised method to generate the positive pool from the corpus, even with some noise.

Acknowledgments

This research was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), National Science Foundation IIS-1320617 and IIS 16-18481, grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov), and a Google PhD Fellowship. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies of the U.S. Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] K. Ahmad, L. Gillam, L. Tostevin, et al., "University of surrey participation in TREC8: Weirdness indexing for logical document extrapolation and retrieval (wilder)," in *Proc. TREC*, pp. 1–8, 1999.

[2] A. Allahverdyan and A. Galstyan, "Comparative analysis of viterbi training and maximum likelihood estimation for HMMs," in *Proc. NIPS*, 2011, pp. 1674–1682.

[3] T. Baldwin and S. N. Kim, "Multiword expressions," *Handbook of Natural Language Processing*, 2nd ed. San Rafael, CA, USA: Morgan and Claypool, 2010.

[4] S. Bedathur, K. Berberich, J. Dittrich, N. Mamoulis, and G. Weikum, "Interesting-phrase mining for ad-hoc text analytics," *Proc. VLDB Endow.*, vol. 3, no. 1/2. pp. 1348–1357, Sep. 2010.

[5] L. Breiman, "Randomizing outputs to increase prediction accuracy," *Mach. Learn.*, vol. 40, no. 3, pp. 229–242, 2000.

[6] K.-H. Chen and H.-H. Chen, "Extracting noun phrases from large-scale texts: A hybrid approach and its automatic evaluation," in *Proc. 32nd Annu. Meeting Assoc. Comput. Linguistics*, 1994, pp. 234–241.

[7] M. Danilevsky, C. Wang, N. Desai, X. Ren, J. Guo, and J. Han, "Automatic construction and ranking of topical keyphrases on collections of short documents," in *Proc. SIAM Int. Conf. Data Mining*, 2014, pp. 398–406.

[8] M.-C. De Marneffe, B. MacCartney, C. D. Manning, et al., "Generating typed dependency parses from phrase structure parses," in *Proc. LREC*, 2006, vol 6, pp. 449–454.

[9] P. Deane, "A nonparametric method for extraction of candidate phrasal terms," in *Proc. 43rd Annu Meeting Assoc. Comput. Linguistics*, 2005, pp. 605–613.

[10] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han, "Scalable topical phrase mining from text corpora," *Proc. VLDB Endow.*, vol. 8, no. 3, pp. 305–316, Nov. 2014.

[11] D. A. Evans and C. Zhai, "Noun-phrase analysis in unrestricted text for information retrieval, "in *Proc. 34th Annu. Meeting Assoc. Comput. Linguistics*, 1996, pp. 17–24.

[12] G. Finch. *Linguistic Terms and Concepts*. New York, NY, USA: Macmillan, 2000.

[13] K. Frantzi, S. Ananiadou, and H. Mima, "Automatic recognition of multi-word terms:. The c-value/nc-value method," *Int. J Digit. Libraries*, vol. 3, no. 3, pp. 115–130, 2000.

[14] C. Gao and S. Michel, "Top-k interesting phrase mining in ad-hoc collections using sequence pattern indexing," in *Proc. 15th Int Conf. Extending Database Technol.*, 2012, pp. 264–275.

[15] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 3, pp. 3–42, 2006.

[16] M. A. Halliday, et al., "Lexis as a linguistic level," in *Memory of JR Firth*, vol. 148, p. 162, 1966.

[17] K. S. Hasan and V. Ng, "Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art," in *Proc. 23rd Int. Conf. Comput. Linguistics: Posters*, 2010, pp. 365–373.

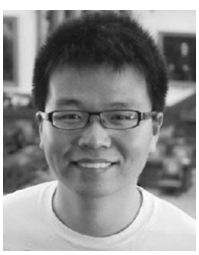
[18] T. Koo, X. Carreras, and M. Collins, "Simple semi-supervised dependency parsing," in *Proc. ACL-HLT*, 2008, pp. 595–603.

[19] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 497–506.

[20] R. Levy and C. Manning, "Is it harder to parse chinese, or the chinese treebank? in *Proc. 41st Annu Meeting Assoc. Comput. Linguistics* - Vol. 1, 2003, pp. 439–446.

[21] B. Li, B. Wang, R. Zhou, X. Yang, and C. Liu, "Citpm: A cluster-based iterative topical phrase mining framework," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2016, pp. 197–213.
[22] B. Li, X. Yang, B. Wang, and W. Cui, "Efficiently mining high quality phrases from texts," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3474–3481.
[23] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han, "Mining quality phrases from massive text corpora," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 1729–1744.
[24] Z. Liu, X. Chen, Y. Zheng, and M. Sun, "Automatic keyphrase extraction by bridging vocabulary gap," in *Proc. 15th Conf. Comput. Natural Lang. Learn.*, 2011, pp. 135–144.
[25] G. Martínez-Muñoz and A. Suárez, "Switching class labels to generate classification ensembles," *Pattern Recognit.*, vol. 38, no. 3, pp. 1483–1494, 2005.
[26] R. McDonald, F. Pereira, K. Ribarov, and J. Hajič, "Non-projective dependency parsing using spanning tree algorithms," in *Proc. Conf. Human Lang. Technol. Empirical Methods Natural Lang. Process.*, 2005, pp. 523–530.
[27] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," in *Proc. 2004 Conf. Empirical Methods Natural Lang. Process.*, 2004.
[28] J. Nivre, M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, et al., "Universal dependencies v1: A multilingual treebank collection," in *Proc. 10th Int. Conf. Lang. Res. Eval.*, 2016, pp. 1659–1666.
[29] P. Deepak, A. Dey, and D. Majumdar, "Fast mining of interesting phrases from subsets of text corpora," in *Proc. 17th Int. Conf. Extending Database Technol*, 2014, pp. 193–204.
[30] A. Parameswaran, H. Garcia-Molina, and A. Rajaraman, "Towards the web of concepts: Extracting concepts from large datasets," *Proc. VLDB Endow.*, vol. 3, no. 1/2, pp. 566–577, Sep. 2010.
[31] Y. Park, R. J. Byrd, and B. K. Boguraev, "Automatic glossary extraction: Beyond terminology identification," in *Proc. 19th Int. Conf. Comput. Linguistics* - Vol. 1, 2002, pp. 1–7.
[32] V. Punyakanok and D. Roth, "The use of classifiers in sequential inference," *Adv. Neural Inf. Process. Syst.*, pp. 995–1001, 2001.
[33] J. Rafiei-Asl and A. Nickabadi, "Tsake: A topical and structural automatic keyphrase extractor," *Appl. Soft Comput.*, vol. 58, pp. 620–630, 2017.
[34] C. Ramisch, A. Villavicencio, and C. Boitet, "Multiword expressions in the wild? the mwetoolkit comes in handy," in *Proc. COLING*, 2010, pp. 57–60.
[35] H. Schmid, "Treetagger—A language independent part-of-speech tagger," *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, vol. 43, 1995, Art. no. 28.
[36] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning , "Kea: Practical automatic keyphrase extraction," in *Proc. 4th ACM Conf. Digit. Libraries*, 1999, pp. 254–255.
[37] E. Xun, C. Huang, and M. Zhou, "A unified statistical model for the identification of english basenp," in *Proc. 38th Annu. Meet. Assoc. Comput. Linguistics*, 2000, pp. 109–116.
[38] Z. Zhang, J. Iria, C. A. Brewster, and F. Ciravegna, "A comparative evaluation of term recognition algorithms," *Proc. Int. Conf. Lang Resources Eval.*, 2008, pp. 2108–2111.

Jingbo Shang working toward the PhD degree in the Department of Computer Science, University of Illinois at Urbana-Champaign. His research focuses on mining and constructing structured knowledge from massive text corpora with minimum human effort. His research has been recognized by many prestigious awards, including the Computer Science Excellence Scholarship from CS@Illinois, Grand Prize of Yelp Dataset Challenge in 2015, and Google PhD Fellowship in Structured Data and Database Management in 2017.



Jialu Liu received the PhD degree from the University of Illinois at Urbana Champaign, in 2015, supervised by Prof. Jiawei Han. He is working with Google Research New York on structured data for knowledge exploration. His primary research interests include scalable information extraction and text mining.

Meng Jiang received the BE and PhD degrees from the Department of Computer Science and Technology, Tsinghua University, in 2010 and 2015, respectively. He is now an assistant professor in the Department of Computer Science and Engineering, University of Notre Dame. He worked as a postdoctoral research associate with the University of Illinois at Urbana-Champaign from 2015 to 2017. He has published more than 20 papers on behavior modeling and information extraction in top conferences and journals of the relevant field such as the *IEEE Transactions on Knowledge and Data Engineering*, ACM SIGKDD, AAAI, ACM CIKM, and IEEE ICDM. He also has delivered six tutorials on the same topics in major conferences. He was the best paper finalist in ACM SIGKDD 2014.

Xiang Ren received the PhD degree from CS@UIUC. He is an assistant professor in the Department of Computer Science, USC. His research develops data-driven and machine learning methods for turning unstructured text data into machine-actionable structures. More broadly, his research interests span data mining, machine learning, and natural language processing, with a focus on making sense of big text data. His research has been recognized with several prestigious awards including a Google PhD Fellowship, a Yahoo!-DAIS Research Excellence Award, a WWW 2017 Outstanding Reviewer Award, a Yelp Dataset Challenge award, and a C. W. Gear Outstanding Graduate Student Award from CS@Illinois. Technologies he developed has been transferred to the US Army Research Lab, National Institute of Health, Microsoft, Yelp, and TripAdvisor.

Clare R. Voss received the BA degree in linguistics from the University of Michigan, the MA degree in psychology from the University of Pennsylvania, and the PhD degree in computer science from the University of Maryland. She is a senior research computer scientist with the Army Research Laboratory (ARL), in Adelphi, Maryland. She has been actively involved in natural language processing (NLP) for over 20 years and continuing as a founding member of the multilingual computing group with ARL, where she now leads an interdisciplinary team working on multilingual and multimodal information extraction in support of event analysis for decision makers, as well as joint navigation and exploration using natural language dialog between humans and robots. She is a member of the Advisory Board for the Computational Linguistics Program at the University of Washington and a past member of the Board of Directors of the Association for Machine Translation in the Americas (AMTA).

Jiawei Han is Abel Bliss professor in the Department of Computer Science, University of Illinois. He has been researching into data mining, information network analysis, and database systems, with more than 600 publications. He served as the founding editor-in-chief of the *ACM Transactions on Knowledge Discovery from Data (TKDD)*. He has received the ACM SIGKDD Innovation Award (2004), IEEE Computer Society Technical Achievement Award (2005), IEEE Computer Society W. Wallace McDowell Award (2009), and Daniel C. Drucker Eminent Faculty Award at UIUC (2011). He is currently the director of the Information Network Academic Research Center (INARC) supported by the Network Science-Collaborative Technology Alliance (NS-CTA) program of the U.S. Army Research Lab. His co-authored textbook *Data Mining: Concepts and Techniques* (Morgan Kaufmann) has been adopted worldwide. He is a fellow of the IEEE and ACM.