Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Incorporating Expert Knowledge into Keyphrase Extraction

Sujatha Das Gollapalli, Xiao-Li Li
Institute for Infocomm Research, A*STAR, Singapore
{gollapallis,xlli}@i2r.a-star.edu.sg

Peng Yang
Tencent AI Lab, Shenzhen, China
yangpeng1985521@gmail.com

Abstract

Keyphrases that efficiently summarize a document's content are used in various document processing and retrieval tasks. Current state-of-the-art techniques for keyphrase extraction operate at a phrase-level and involve scoring candidate phrases based on features of their component words. In this paper, we learn keyphrase taggers for research papers using token-based features incorporating linguistic, surface-form, and document-structure information through *sequence labeling*. We experimentally illustrate that using within-document features alone, our tagger trained with Conditional Random Fields performs on-par with existing state-of-the-art systems that rely on information from Wikipedia and citation networks. In addition, we are also able to harness recent work on *feature labeling* to seamlessly incorporate expert knowledge and predictions from existing systems to enhance the extraction performance further. We highlight the modeling advantages of our keyphrase taggers and show significant performance improvements on two recently-compiled datasets of keyphrases from Computer Science research papers.

Introduction

Keyphrases (or keywords) that provide a concise representation of the topical content of a document are used in various data mining and web-related tasks (Hammouda, Matute, and Kamel 2005; Bao et al. 2007; Xu et al. 2008; Li et al. 2010). *Keyphrase extraction*, the challenging task of automatically extracting a small set of representative keyphrases continues to garner research interest in AI and natural language processing (NLP) communities (Wan and Xiao 2008; Hasan and Ng 2010).

Various supervised and unsupervised techniques are available for keyphrase extraction (Hasan and Ng 2014). Most state-of-the-art systems first extract a set of candidate phrases for a given document during keyphrase extraction (Frank et al. 1999; Medelyan, Frank, and Witten 2009; Gollapalli and Caragea 2014). These systems employ phrase-filtering on the set of all n-grams of an input document to remove phrases that are unlikely to be human-generated keyphrases. For instance, n-grams ending in stopwords or *prepositions* are unlikely to be author-specified keyphrases. Linguistic filters effectively reduce the number of possible n-grams that need to be considered by subsequent scoring and classification modules. Typically, the value of n is set to $\{1,2,3\}$ based on the observation that author-specified keyphrases tend to be uni/bi/tri-grams in practice (Caragea et al. 2014).

In unsupervised models, candidate phrases are scored based on individual tokens comprising them.[1] Various "goodness" or "interestingness" measures that reflect document-level, corpus-level, and external statistics are used in this scoring (Hasan and Ng 2010). For example, the TextRank algorithm builds a graph based on neighboring words in a document and computes the score of each word as the PageRank centrality measure of its corresponding node in the word graph (Mihalcea and Tarau 2004).

In contrast, supervised models use known ("correct") keyphrases to frame keyphrase identification as a binary classification task. Candidate phrases from the training set of documents are assigned positive and negative labels and features such as part-of-speech (POS) tags, phrase length, occurrence frequency, and position information in the document are used for learning keyphrase classifiers (Hasan and Ng 2014). Ranking approaches which use ordering information among candidate phrases to train extraction models were also investigated previously (Jiang, Hu, and Li 2009).

In this paper, we avoid the candidate phrase extraction step by formulating keyphrase extraction as a *sequence tagging/labeling* task. Given a stream of tokens corresponding to the content of a document,[2] a keyphrase tagger assigns to each token position a tag/label from the set $\{$KP, O$\}$ where the label **KP** corresponds to a keyphrase token and **O** refers to a non-keyphrase token. An example is shown in Table 1. Unlike phrase-based approaches where candidate phrases comprise (multiple) training/test instances for a document, the entire content of the document comprises a single instance for a sequence tagging model.

The example in Table 1 refers to the title of a research paper published in the World Wide Web conference in the year 2010 and is part of the recently-compiled datasets for keyphrase extraction described further in the **Experiments** section. We highlight some shortcomings of existing sys-

[1] We use "token" and "word" interchangeably.

[2] We assume textual content and whitespace tokenization.

Tokens:	Visualizing	differences	in	web	search	algorithms	using	the	**expected**	**weighted**	**Hoeffding**	**distance**
POS tags:	VBG	NNS	IN	NN	NN	VBZ	VBG	DT	VBN	JJ	VBG	NN
Phrase tags:	NP	NP	PP	NP	NP	VP	VP	NP	NP	NP	NP	NP
Labels:	O	O	O	O	O	O	O	O	KP	KP	KP	KP

Table 1: (Example) The title of a research paper is shown with its tokens, POS and phrase tags, and keyphrase labels.

tems in handling this example.

Several keyphrase extraction algorithms including the recent ExpandRank, CiteTextRank, and CeKE systems (Wan and Xiao 2008; Gollapalli and Caragea 2014; Caragea et al. 2014) employ part-of-speech criteria during phrase filtering. Specifically, these systems only consider phrases comprising of nouns and adjectives with POS tags from the set {NN, NNS, NNP, NNPS, JJ} for scoring.[3] In addition, based on the value for n during n-gram generation, these systems may not generate candidate phrases with more than three tokens. In Table 1, the author-specified keyphrase highlighted in **bold** has four words as well as POS tags referring to verbs[4] and is automatically excluded from consideration by several existing systems (Hasan and Ng 2014).

Indeed, we noticed that about 8% of author-specified keyphrases in our experimental datasets have tags other than nouns and adjectives and about 1% of them have more than three tokens. The keyphrase extraction algorithm based on sequence tagging described in this paper does not involve an explicit phrase extraction step and is able to consider all possible candidate phrases of any arbitrary length by default. Although concerns related to missing phrases in classifiers may be addressed by including all possible n-grams, in practice, such an inclusion results in several noisy phrases that affect learning algorithms.

Incorporating Expert Knowledge: Several recent state-of-the-art keyphrase extraction systems incorporate external sources of evidence and domain-specific knowledge along with document and corpus-level information while scoring candidate keyphrases. For example, Maui uses semantic information based on Wikipedia (Medelyan, Frank, and Witten 2009) whereas the CeKE system (Caragea et al. 2014) includes features based on the document-citation network obtained from CiteSeer^x (Li et al. 2006).

In most existing systems specialized knowledge is incorporated into the extraction process through complex features such as "the position of the first occurrence of a phrase divided by the total number of tokens" (Frank et al. 1999; Hulth 2003), "the distribution of terms among different document sections" (Nguyen and Kan 2007), "the distance of the first occurrence of a phrase from the beginning of a paper is below some value β" (Caragea et al. 2014), and "number of links to the Wikipedia article referring to the phrase" (Medelyan, Frank, and Witten 2009).

In lieu of intricate features such as the above, we harness the recent work on weak supervision to specify expert hints and external knowledge during keyphrase extraction through simple label-probability distributions. For example, specifying expert knowledge that "a noun word that occurs in the document's title is more likely to be keyphrase" involves simply expressing a label-distribution preference with the corresponding feature as ("isInTitleAndNoun" {KP:0.9 O:0.1}). We use label distributions to incorporate expert hints into model training through the *posterior regularization* framework (Mann and McCallum 2008). Our contributions in this paper are listed below:

1. We study keyphrase extraction as a sequence tagging task and design features for learning a keyphrase tagger using Conditional Random Fields or CRFs (Lafferty, McCallum, and Pereira 2001). In contrast with several existing works, our set of features is minimalistic with all features representing linguistic, orthographic, and structure information extracted from within the document.
2. We investigate feature-labeling and posterior regularization as a means to seamlessly integrate expert-knowledge and domain-specific hints during keyphrase extraction. To the best of our knowledge, we are the first to study weak supervision as an alternative to intricate feature design to achieve this objective.
3. We illustrate the performance of our keyphrase taggers on two recently-compiled datasets of research papers in Computer Science. Our models are able to perform on-par with several state-of-the-art systems that make use of external evidence from citations and Wikipedia despite only using within-document features. Additionally, when external evidence is incorporated through feature labeling, we *significantly out-perform* existing baselines on both the datasets. We show performance benefits with both expert-specified and automatically-generated labeled features on our experimental datasets.

We summarize the features used to train our keyphrase taggers and introduce the feature labeling framework in the next section (**Proposed Methods**). Datasets, baselines and the experimental setup used to evaluate our models are described in the **Experiments** section whereas closely-related recent work is briefly summarized in the **Related Work** section. Finally, we conclude the paper with a summary and notes on future directions.

Proposed Methods

Sequence tagging involves the prediction of a sequence of labels $\mathbf{y} = <y_1 \dots y_N>$ given an input sequence of tokens: $\mathbf{t} = <t_1 \dots t_N>$ (Sarawagi 2005). Each position $i : 1 \dots N$ in the input sequence of tokens can be modeled by vectors of features $<\mathbf{x}_1 \dots \mathbf{x}_N>$. Although various generative and discriminative models exist for learning sequence taggers, Conditional Random Fields were shown to obtain state-of-the-art performance on several IE and NLP related tagging

[3]The Penn Treebank list of tags is available at: https://catalog.ldc.upenn.edu/docs/LDC99T42/tagguid1.pdf.

[4]We obtain the POS and phrase tags using the Stanford parser.

tasks that involve several complex, interdependent features (Sutton and McCallum 2012).

Features for Keyphrase Tagging

We train a keyphrase tagger using CRFs with the following three types of features.

1. **Word, orthographic, and stopword features**: We use whitespace tokenization, convert all tokens to lowercase after removing punctuation and use the stemmed form corresponding to the token (obtained using the Porter stemmer (1997)) for word features. We add a special feature "allPunct" to capture tokens only comprised of punctuation as well as boolean features "isCapitalized" and "isStopword" to indicate if the word is capitalized or a stopword.[5] In addition, the end of a sentence is explicitly indicated using an "EOL" feature to capture sentence boundary information.
2. **Parse-tree features**: We obtain the lexicalized parse of the document content using the case and punctuation cues provided by the author of the document. The Stanford Parser (Finkel, Grenager, and Manning 2005)[6] was used to obtain the level-1 and level-2 parse tags comprising the part-of-speech (POS) and phrase features at each word position. Hulth (2003) showed that incorporating linguistic knowledge such as NP-chunking and POS tags dramatically improves extraction performance over using statistical features alone and almost all existing works incorporate POS tags in their models (Hasan and Ng 2014).
3. **Title features**: We indicate if a non-stopword is part of the document's title using a boolean feature ("isInTitle"). The title of a document can be considered a summary sentence describing the document and authors often add discriminative words in their titles. The *isInTitle* feature depends on document structure information that is often part of research paper datasets (Kim et al. 2010).

Given the token stream corresponding to a document, let F, G represent feature-types described above (word, POS etc.) and i represent a token position. The feature templates used for training our keyphrase tagger are listed below:

Unigram features	F_i
Bigram features	$F_{i-1}F_i$ and F_iF_{i+1}
Skipgram features	$F_{i-1}F_{i+1}$
Compound features	F_iG_i

Unigram features refer to the features generated at position i using the token at that position (e.g., POS tag for the word at position i). The neighborhood information for a given position is incorporated using the bigram and skipgram features that reference tokens at the previous and next positions relative to i. Intuitively, if a current token is part of a multiterm phrase, this may be indicated via suitable bigram and skipgram features (e.g., they may share the same phrase tags). Compound features are conjunctions combining features at a given position. For example, the feature "isInTitle and POS=NN" is stronger evidence to the tagger than each of these features in isolation.

Illustrative Example: A partial list of features extracted for the token "expected" from the anecdotal example in Table 1 are shown in Table 2 for illustration. The unigram features comprise of the stemmed token, the POS and phrase tags, and boolean features indicating the lack of capitalization, presence in the title as well as a feature indicating that this token is not a stopword. The "big1" and "big-1" prefixes indicate bigrams involving the current token with its next and previous token positions respectively. For example, the feature "big1_notStopword_notStopword" captures the information that both the current token ("expected") as well as the next token ("weighted") are not stopwords. The "cmpd-L1-VBN_isInTitle" feature captures the information that the token is both a verb as well as in the title, while "skip-1-L1-DT_L1-JJ" captures the adjacent POS tag features of the tokens "the" and "weighted" respectively.

We train our CRF tagger using unigram features corresponding to all feature types and bigram, skipgram, and compound features corresponding to orthographic, stopword, parse-tree, and title features. In experiments, we commonly refer to "bigram, skipgram, and compound features" as *neighborhood* features. Note that in contrast with some of the intricate features mentioned in the **Introduction** section, our features are fairly simple in design and are also commonly employed in other IE and NLP tagging tasks (Sarawagi 2005; Indurkhya and Damerau 2010).

Baselines: We compare our tagger with recent state-of-the-art systems: Kea, Maui, and CeKE. The Kea system originally proposed in (Frank et al. 1999) has been significantly enhanced since and forms a competitive baseline using document, thesaurus, and corpus-based features such as TFIDF, length of the phrase, and first occurrence. Maui augments features in Kea with several novel features such as spread of the phrase and keyphraseness. In addition, phrases are mapped to specific Wikipedia article pages and features such as node-degree of the page in the Wikipedia graph and occurrence of the phrase in the link of the page are used in Maui (Medelyan, Frank, and Witten 2009). The CeKE system, designed for research papers, augments features from Kea with several additional features based on term occurrence in citation contexts and citation-based TFIDF (Caragea et al. 2014).

All the above baseline systems are phrase-based supervised techniques and have publicly-available implementations. Unlike Kea, Maui and CeKE use external evidence from Wikipedia and citation network respectively. To the best of our knowledge, these systems comprise the most recent algorithms involving supervised techniques. Additionally, for the research paper datasets used in this paper, CeKE was shown to outperform both Kea and the TextRank-family of unsupervised techniques (Caragea et al. 2014).

Feature Labeling and Posterior Regularization

Mann, Druck and McCallum proposed the feature labeling framework as a means to incorporate expert-provided hints into CRF-based taggers (2008). For example, to capture the expert intuition that *noun words occurring in the paper titles*

[5] We used the stopword list from Maui (Medelyan, Frank, and Witten 2009).

[6] http://nlp.stanford.edu/software/lex-parser.shtml

Type of feature	Partial list of features
Unigrams	expect, L1-VBN, L2-NP, isInTitle, notCapitalized, notStopword
Bigrams	big1_notStopword_notStopword, big-1-L1-DT_L1-VBN, big-1-L2-NP_L2-NP
Skipgrams	skip-1-L1-DT_L1-JJ, skip-1-L2-NP_L2-NP , skip-1-isStopword_notStopword
Compounds	cmpd-L1-VBN_L2-NP, cmpd-L1-VBN_isInTitle, cpmd-L1-VBN_notStopword

Table 2: Sample features are shown for the token "expected" in the example from Table 1.

are more likely to be keyphrases, the feature "cmpd-L1-NN-isInTitle" (using our previous notation) can be assigned a label distribution: {KP: 0.9 O:0.1} indicating a preference for tokens with the above feature to be marked with the label 'KP' with very high probability (90% of the time). Unlike standard CRF models that are trained on fully-annotated training instances (supervised models), the Posterior Regularization (PR) framework incorporates information from individual labeled features such as the above into the CRF parameter estimation process thus allowing for "weak supervision" into the learning process.

In the PR framework, the specified feature-label distributions are converted to a set of linear constraints on model posterior expectations for the features. The objective function of the CRF is suitably modified to include an additional factor capturing the KL-divergence between posteriors based on labeled features and the original model-estimated posteriors for the same features (Mann and McCallum 2010; Ganchev et al. 2010). Mann and McCallum showed that given limited annotation time, expert-specified labeled features can be used to improve discriminative models over other semi-supervised approaches that use fully-labeled instances. In addition, given sufficient annotated data, various techniques were studied to automatically estimate feature-label distributions for specific tagging problems (Haghighi and Klein 2006; Druck, Mann, and McCallum 2008; Gollapalli et al. 2014).

The standard model training process in CRF also estimates label distributions for features based on the (feature, label) co-occurrence counts in training data during likelihood computation (Sutton and McCallum 2012). Labeled features are useful when such an estimation is not accurately possible due to lack of sufficient number of training instances capturing (feature, label) co-occurrence. Consequently, this information when specified explicitly using an 'expert' label-distribution (e.g., {KP:0.9 O:0.1}) comprises additional information for the learning algorithm.

Expert Features: We capture observations based on previous research in keyphrase extraction using three sets of "expert" labeled features and enforce them using the posterior regularization framework in experiments.

The first set of features in Table 3 captures predictions from baseline systems CeKE and Maui (described in the previous section). That is, when a given phrase is identified by known supervised techniques (CeKE and Maui), we indicate the high likelihood of this word indeed being a keyphrase through the corresponding labeled features. Thus, we fold in predictions from phrase-based classifiers within the tagger in a two-step setting with this set of labeled features.

The second set of features captures preferences for noun phrases occurring in document titles and citation contexts. Previous studies related to keyphrase extraction in research papers have found these features to be highly indicative of keyphrases (Kim et al. 2010; Gollapalli and Caragea 2014).

Set 1 (Predictions from Phrase-based Classifiers)		
cmpd-CeKEKP_MauiKP	KP:0.9	O:0.1
CeKEKP	KP:0.8	O:0.2
MauiKP	KP:0.8	O:0.2
Set 2 (Presence in Document Title and Citation Contexts)		
cmpd-L2-NP_isInCitingContexts	KP:0.8	O:0.2
cmpd-L2-NP_isInCitedContexts	KP:0.8	O:0.2
cmpd-L2-NP_isInTitle	KP:0.8	O:0.2
Set 3 (Predictions from Unsupervised Models)		
cmpd-L2-NP_OneUKP	KP:0.7	O:0.3
cmpd-L2-NP_TwoUKP	KP:0.8	O:0.2
cmpd-L2-NP_AllUKP	KP:0.9	O:0.1

Table 3: Sample 'expert' features and label distributions

Finally, for the third set of labeled features, we incorporate information from existing unsupervised keyphrase (UKP) extraction algorithms: TFIDF, TextRank, SingleRank, and ExpandRank (Mihalcea and Tarau 2004; Wan and Xiao 2008). We indicate preferences for words in noun-phrases that were marked among the top-10 predictions from at least one, two and all the unsupervised methods indicated by OneUKP, TwoUKP, and AllUKP respectively.

Labeled Features through Feature Selection: We also study automatic techniques to extract labeled features by applying standard feature selection measures on instances in the training data. Similar to prior works (Haghighi and Klein 2006; Druck, Mann, and McCallum 2008; Gollapalli et al. 2014), we extract the features that co-occur with the 'KP' label with larger than average frequency. The **autoPMI** list refers to features having larger Pointwise Mutual Information with the label 'KP' than with the label 'O' ranked based on PMI values whereas **autoFreq** refers to features ranked based on their occurrence frequency. The top-10 features from these two rankings are assigned a heuristic label distribution {KP:0.9 O:0.1} to form labeled feature sets.

Experiments

Datasets

We evaluate our models using the research paper datasets collected by recent works on keyphrase extraction (Gollapalli and Caragea 2014). To the best of our knowledge, these datasets comprise the largest, publicly-available benchmark datasets of research paper abstracts containing both author-specified keyphrases and citation network information. Abstracts from these datasets are from papers published in two

Venue	**#Abs/#KPs** (Org)	**#Abs/#KPs** (Locatable)	**Number of keyphrases with different lengths**
KDD	365/1467	315/717	{#unigrams 221, #bigrams 404, #trigrams 80, #>trigrams 12}
WWW	425/2065	388/905	{#unigrams 368, #bigrams 451, #trigrams 79, #>trigrams 7}

Table 4: Summary of Datasets. The total numbers of abstracts and keyphrases in the original dataset are shown with the numbers of abstracts for which at least one author-specified keyphrase could be located along with the total number of keyphrases located.

premier conferences: the World Wide Web (WWW) Conference and the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). The incoming and outgoing citation contexts for each paper were obtained from CiteSeer^x, the digital library portal for Computer Science related literature (Li et al. 2006).

Similar to previous works, we evaluate the predictions of each extraction algorithm against the author-specified keyphrases that can be located in the corresponding paper abstracts in the dataset ("gold standard"). We employ 10-fold cross-validation and present (micro) averaged results for all our experiments using the precision, recall, and F1 measures. For comparing two methods, we choose the F1 measure that represents a balance between precision and recall (Manning, Raghavan, and Schütze 2008).

The datasets are summarized in Table 4 along with the number of keyphrases originally specified with each paper and the number of keyphrases locatable in the paper abstracts.[7] We also indicate the number of keyphrases with one, two, three, and more than three tokens found in these abstracts. As observed previously (Caragea et al. 2014), very few (only about 1%) author-specified keyphrases have greater than three tokens. We found that about 8-9% of the gold keyphrases do not satisfy the noun or adjective POS filters used in previous works.

We used the CRF and posterior regularization implementations provided as part of the Mallet toolkit (McCallum 2002). Default parameter settings were used while training the standard CRF models. For posterior regularization, we set the constraint weights to 50 and the number of iterations for the EM-style optimization algorithm to 100.[8] Publicly-available implementations for Kea,[9] Maui,[10] and CeKE[11] were used in baseline experiments.[12]

Results and Discussion

Tagging compared with baselines The results of ten-fold cross-validation experiments with CRF taggers trained using unigram and neighborhood features are compared with the baselines in Figure 1 (a). Our CRF taggers significantly outperform Kea which only use document and corpus-level features and also perform on-par with CeKE and Maui that incorporate features from CiteSeer^x and Wikipedia respectively. We did not directly use the CeKE numbers from (Caragea et al. 2014) since they are only computed on phrases that satisfy the POS filters mentioned in the **Introduction** section. Based on the results from this table, we conclude that sequence tagging models are more effective for keyphrase extraction than phrase-based classifiers. However, phrase-based models obtain higher recall compared to the CRF taggers that achieve higher precision and an overall better F1 score. In a later experiment, we incorporate predictions from phrase-based models using Set-1 labeled features and PR to further improve extraction performance.

Effect of neighborhood and title features The ten-fold CV tagging performance is shown on the WWW dataset using unigram features (**UF**) alone, both unigram and neighborhood features (**UF+NF**), and without the title features (**noTitle**) in Figure 1 (b). When predicting tags at a given token position, neighborhood information incorporated via bigrams, skipgrams, and conjunctions are effectively harnessed via edge-transition parameters (Sutton and McCallum 2012) in a CRF resulting in better performance with UF+NF features. Similarly, as seen in the noticeable dip in the performance measures when title-based features are excluded, we conclude that content words present in the titles of research papers are very likely to be part of keyphrases.

Performance with expert-labeled features For these experiments, we first train the standard CRF tagger on the train split of data (as before). Next, posterior regularization is applied on the test instances in transductive mode (Bishop 2006). The extraction performance using the different sets of expert labeled features listed in Table 3 as well as the labeled features extracted automatically (autoFreq and autoPMI settings) are shown in Figure 1 (a). As the numbers indicate, the PR framework is extremely effective in incorporating the external knowledge specified as labeled features into the model estimation process for both the datasets. All sets of labeled features including the automatically extracted ones result in performance benefits over using CRF alone. In particular, the best performance (rows marked in **bold**) is obtained with Set-1 that incorporates predictions from CeKE and Maui.

In Figure 2 (c), we illustrate PR with Set-1 for the WWW dataset. By employing predictions from Maui and CeKE as labeled features (**CRF+Set-1 w PR**), we are able to do significantly better than both these systems (Maui is the best performing baseline method on the WWW dataset) as well as the original CRF tagger. Specifically, we are able to improve the tagger's recall highlighted in the previous discussion. The **CRF+Set-1** bars indicate tagging performance when CeKE and Maui predictions are incorporated as regular features in the CRF. Note that these additional features yield no additional performance benefits. As described previously, a potential reason for this behaviour is the lack of sufficient evidence in the training data for accu-

[7] Some gold keyphrases probably part of fulltext are missing from the abstracts.

[8] http://mallet.cs.umass.edu/semi-sup-fst.php

[9] http://www.nzdl.org/Kea/index.html

[10] https://github.com/zelandiya/maui

[11] http://www.cse.unt.edu/∼ccaragea/keyphrases.html

[12] Processed datasets and code are available upon request.

(a)

Setting	KDD Precision	KDD Recall	KDD F1	WWW Precision	WWW Recall	WWW F1
Kea	0.1551	0.3278	0.2105	0.1549	0.3182	0.2084
CeKE	0.2174	**0.3905**	0.2793	0.2251	0.2519	0.2377
Maui	0.1695	0.3724	0.2329	0.1837	**0.3934**	0.2504
CRF	**0.4068**	0.2162	**0.2823**	**0.3689**	0.1945	**0.2547**
CRF with labeled features and PR						
Set 1	0.2777	**0.4407**	**0.3407**	0.2668	**0.3558**	**0.3049**
Set 2	0.3130	0.3013	0.3070	0.2995	0.2575	0.2768
Set 3	0.3091	0.3026	0.3058	0.2969	0.2674	0.2814
Top-10 PMI	0.2469	0.4114	0.3085*	0.2511	0.3282	0.2845*
Top-10 Frequency	**0.3282**	0.2649	0.2932	**0.3323**	0.2298	0.2717

(b)

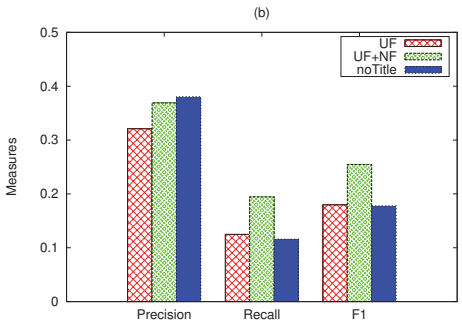

Figure 1: (a) Ten-fold CV performance of the baseline methods, CRF, and posterior regularization; (b) Performance of the CRF tagger with different feature sets on the WWW dataset

(a)

Top-10 features (PMI)
big-1-isInTitle_isInTitle
cmpd-isInTitle_isCapitalized
cmpd-isInTitle_oneUKP
cmpd-isInTitle_AllUKP
cmpd-isInTitle_TwoUKP
big1-isInTitle_isInTitle
cmpd-L2-NP_isInTitle
cmpd-L1-NN_isInTitle
cmpd-isInTitle_isInCitingContext
cmpd-isInTitle_isInCitedContext

(b)

Top-10 features (Frequency)
big-1-L2-NP_L2-NP
big1-isInCitedContext_isInCitedContext
big1-isInCitingContext_isInCitingContext
big1-L2-NP_L2-NP
big-1-isInCitedContext_isInCitedContext
big-1-isInCitingContext_isInCitingContext
cmpd-isInCitingContext_isInCitedContext
cmpd-L2-NP_isInCitingContext
cmpd-L2-NP_isInCitedContext
cmpd-isInTitle_isInCitingContext

(c)

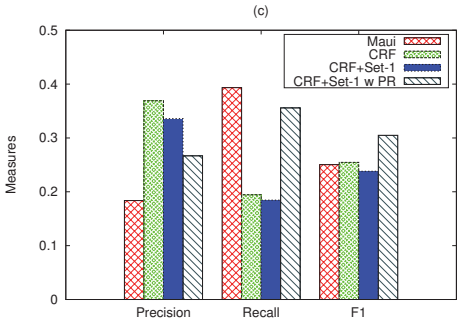

Figure 2: (a) and (b) Top-10 features based on PMI and frequency; (c) Ten-fold CV tagging performance with PR (WWW dataset)

rate model parameter estimation thus requiring explicit preference specification via labeled features.

Performance with automatically-extracted labeled features We incorporate 'isInCitedContexts', 'isInCitingContexts', 'OneUKP', 'TwoUKP' and 'AllUKP' as regular features into the training data and apply feature selection using frequency and PMI information. The top-10 features extracted by these methods are assigned the heuristic distribution '{KP:0.9 O:0.1}' to form labeled features for the autoFreq and autoPMI runs respectively. The top-10 features extracted from the WWW dataset using this process are shown in Figures 2 (a) and (b).

From the results in Figure 1 (a), it can be seen that PMI-based features are better performing of the two sets and also among all sets other than Set-1 (rows marked with '*'). The automatically-extracted PMI features in Figure 2 (a) also make intuitive sense despite no 'expert' guidance.

We note that the observations and trends shown with the WWW datasets also hold for the KDD dataset the plots of which are not included due to space limitations.

Anecdotes Our best-performing models correctly identified 37% of the gold keyphrases with more than three tokens from the experimental datasets. Examples of these keyphrases that also include POS tags not considered in phrase filtering based approaches are "*learning to rank relational objects*" (L1-NNP L1-TO L1-VB L1-JJ L1-NNS), "*end-user quality of experience*" (L1-JJ L1-NN L1-IN L1-NN), and "*quadratically constrained quadratic programming*" (L1-RB L1-VBN L1-JJ L1-NN).

Related Work

Keyphrase extraction is widely studied in various domains (Frank et al. 1999; Kim et al. 2010; Bong and Hwang 2011), for different document-types (Liu et al. 2009; Marujo et al. 2013) and for tag recommendation (Bao et al. 2007; Xu et al. 2008). Supervised techniques for keyphrase extraction are often phrase-based models trained using document and corpus-level features such as POS tags, position of the word and TFIDF information (Frank et al. 1999; Witten et al. 1999; Turney 2000; Hulth 2003). Recent systems also incorporate external features based on citation networks as well as Wikipedia into keyphrase extraction (Caragea et al. 2014; Medelyan, Frank, and Witten 2009). In contrast, several unsupervised extraction techniques score keyphrases based on "goodness" measures of words comprising them using graphs constructed from documents (Mihalcea and Tarau 2004; Boudin 2013; Gollapalli and Caragea 2014; Wang, Liu, and McDonald 2015).

Bhaskar et al. (2012) employ CRFs trained on features such as word presence in document sections such as abstract and title as well as linguistic features such as POS, chunking, and named-entity tags for keyphrase extraction in sci-

entific articles. Similar features were employed by Zhang et al. for documents in Chinese (2008). We investigated CRFs for their modeling advantages as well as their ability to incorporate expert knowledge via weak supervision. Weak supervision was previously studied for several classification, and information extraction problems using techniques such as feature labeling (Haghighi and Klein 2006; Druck, Mann, and McCallum 2008) and based on known knowledge-bases (Hoffmann et al. 2011).

Conclusions

We studied keyphrase extraction as a tagging task with Conditional Random Fields using simple token, parse, and orthographic features. We showed experimentally that CRFs show both modeling and performance advantages over the current state-of-the-art, phrase-based models on research paper datasets. In addition, we are able to incorporate domain knowledge into the extraction process via the feature labeling framework for CRFs to further enhance extraction performance. In future, we would like to explore weak supervision for other types of documents (such as news articles and product reviews) as well as in parallel corpora (Arcan et al. 2014).

Acknowledgments We are grateful to the reviewers for their help in improving the presentation of this paper and to fellow researchers who provided their keyphrase extraction systems and datasets for comparative evaluation.

References

Arcan, M.; Turchi, M.; Tonelli, S.; and Buitelaar, P. 2014. Enhancing statistical machine translation with bilingual terminology in a cat environment. In *Proceedings of the Eleventh Biennial Conference of the Association for Machine Translation in the Americas*.

Bao, S.; Xue, G.; Wu, X.; Yu, Y.; Fei, B.; and Su, Z. 2007. Optimizing web search using social annotations. In *WWW*.

Bhaskar, P.; Nongmeikapam, K.; and Bandyopadhyay, S. 2012. Keyphrase extraction in scientific articles: A supervised approach. In *COLING*.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc.

Bong, S.-Y., and Hwang, K.-B. 2011. Keyphrase extraction in biomedical publications using mesh and intraphrase word co-occurrence information. In *Proceedings of the ACM Fifth International Workshop on Data and Text Mining in Biomedical Informatics*.

Boudin, F. 2013. A comparison of centrality measures for graph-based keyphrase extraction. In *IJCNLP*.

Caragea, C.; Bulgarov, F. A.; Godea, A.; and Gollapalli, S. D. 2014. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *EMNLP*.

Druck, G.; Mann, G.; and McCallum, A. 2008. Learning from labeled features using generalized expectation criteria. In *SIGIR*.

Finkel, J. R.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*.

Frank, E.; Paynter, G. W.; Witten, I. H.; Gutwin, C.; and Nevill-Manning, C. G. 1999. Domain-specific keyphrase extraction. In *IJCAI*.

Ganchev, K.; Graça, J. a.; Gillenwater, J.; and Taskar, B. 2010. Posterior regularization for structured latent variable models. *JMLR*.

Gollapalli, S. D., and Caragea, C. 2014. Extracting keyphrases from research papers using citation networks. In *AAAI*.

Gollapalli, S. D.; Qi, Y.; Mitra, P.; and Giles, C. L. 2014. Extracting researcher metadata with labeled features. In *SDM*.

Haghighi, A., and Klein, D. 2006. Prototype-driven learning for sequence models. In *HLT-NAACL*.

Hammouda, K. M.; Matute, D. N.; and Kamel, M. S. 2005. Corephrase: Keyphrase extraction for document clustering. In *Machine Learning and Data Mining in Pattern Recognition*.

Hasan, K. S., and Ng, V. 2010. Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art. In *COLING*.

Hasan, K. S., and Ng, V. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *ACL*.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *HLT*.

Hulth, A. 2003. Improved automatic keyword extraction given more linguistic knowledge. *EMNLP*.

Indurkhya, N., and Damerau, F. J. 2010. *Handbook of Natural Language Processing*. Chapman & Hall/CRC, 2nd edition.

Jiang, X.; Hu, Y.; and Li, H. 2009. A ranking approach to keyphrase extraction. In *SIGIR*.

Kim, S. N.; Medelyan, O.; Kan, M.-Y.; and Baldwin, T. 2010. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *SemEval*.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Li, H.; Councill, I. G.; Bolelli, L.; Zhou, D.; Song, Y.; Lee, W.-C.; Sivasubramaniam, A.; and Giles, C. L. 2006. Citeseerx: A scalable autonomous scientific digital library. In *Proceedings of the 1st International Conference on Scalable Information Systems*.

Li, Z.; Zhou, D.; Juan, Y.-F.; and Han, J. 2010. Keyword extraction for social snippets. In *WWW*.

Liu, F.; Pennell, D.; Liu, F.; and Liu, Y. 2009. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *NAACL*.

Mann, S. G., and McCallum, A. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of ACL-08: HLT*.

Mann, G. S., and McCallum, A. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *J. Mach. Learn. Res.*

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Marujo, L.; Ribeiro, R.; de Matos, D. M.; Neto, J. P.; Gershman, A.; and Carbonell, J. G. 2013. Key phrase extraction of lightly filtered broadcast news. *CoRR*.

McCallum, A. K. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Medelyan, O.; Frank, E.; and Witten, I. H. 2009. Human-competitive tagging using automatic keyphrase extraction. In *EMNLP*.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into text. In *EMNLP*.

Nguyen, T. D., and Kan, M.-Y. 2007. Keyphrase extraction in scientific publications. In *Proceedings of the 10th International Conference on Asian Digital Libraries: Looking Back 10 Years and Forging New Frontiers*.

Porter, M. F. 1997. Readings in information retrieval. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. chapter An Algorithm for Suffix Stripping.

Sarawagi, S. 2005. *Advanced Methods for Knowledge Discovery from Complex Data*. chapter Sequence Data Mining.

Sutton, C., and McCallum, A. 2012. An introduction to conditional random fields. *Found. Trends Mach. Learn.*

Turney, P. D. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval* 2(4).

Wan, X., and Xiao, J. 2008. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*.

Wang, R.; Liu, W.; and McDonald, C. 2015. Corpus-independent generic keyphrase extraction using word embedding vectors. In *Deep Learning for Web Search and Data Mining*.

Witten, I. H.; Paynter, G. W.; Frank, E.; Gutwin, C.; and Nevill-Manning, C. G. 1999. Kea: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*.

Xu, S.; Bao, S.; Fei, B.; Su, Z.; and Yu, Y. 2008. Exploring folksonomy for personalized search. In *SIGIR*.

Zhang, C.; Wang, H.; Liu, Y.; Wu, D.; Liao, Y.; and Wang, B. 2008. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems* 4(3).