Unsupervised Concept Categorization and Extraction from Scientific Document Titles

Adit Krishnan*, Aravind Sankar*, Shi Zhi, Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign, USA
{aditk2,asankar3,shizhi2,hanj}@illinois.edu

ABSTRACT
This paper studies the automated categorization and extraction of scientific concepts from titles of scientific articles, in order to gain a deeper understanding of their key contributions and facilitate the construction of a generic academic knowledgebase. Towards this goal, we propose an unsupervised, domain-independent, and scalable two-phase algorithm to type and extract key concept mentions into aspects of interest (e.g., Techniques, Applications, etc.). In the first phase of our algorithm we propose PhraseType, a probabilistic generative model which exploits textual features and limited POS tags to broadly segment text snippets into aspect-typed phrases. We extend this model to simultaneously learn aspect-specific features and identify academic domains in multi-domain corpora, since the two tasks mutually enhance each other. In the second phase, we propose an approach based on adaptor grammars to extract fine grained concept mentions from the aspect-typed phrases without the need for any external resources or human effort, in a purely data-driven manner. We apply our technique to study literature from diverse scientific domains and show significant gains over state-of-the-art concept extraction techniques. We also present a qualitative analysis of the results obtained.

KEYWORDS
Concept extraction, Probabilistic model, Adaptor grammar

1 INTRODUCTION
In recent times, scientific communities have witnessed dramatic growth in the volume of published literature. This presents the unique opportunity to study the evolution of scientific concepts in the literature, and understand the contributions of scientific articles via their key aspects, such as techniques and applications studied by them. The extracted information could be used to build a general-purpose scientific knowledgebase which can impact a wide range of applications such as discovery of related work, citation recommendation, co-authorship prediction and temporal evolution of scientific domains. For instance, construction of a Technique-Application knowledgebase can help answer questions such as -

“What methods were developed to solve a particular problem?” and “What were the most popular interdisciplinary techniques or applications in 2016?”.

To achieve these objectives, it is necessary to accurately type and extract the key concept mentions that are representative of a scientific article. Titles of publications are often structured to emphasize their most significant contributions. They provide a concise, yet accurate representation of the key concepts studied. Preliminary analysis of a sample from popular computer science venues in the years 1970-2016 indicates that 81% of all research titles contain at least two concept mentions, where 73% of these titles state both techniques and applications and the remaining 27% contain one of the two aspects. Although a minority may be uninformative, our typing and extraction framework generalizes well to their abstract or introduction texts.

Our problem fundamentally differs from classic Named Entity Recognition techniques which focus on natural language text [17] and web resources via distant supervision [19]. Entity phrases corresponding to predefined categories such as person, organization, location etc are detected using trigger words (pvt., corp., ltd., Mr./Mrs. etc.), grammar properties, syntactic structures such as dependency parses, part-of-speech (POS) tagging and textual patterns. In contrast, academic concepts are not associated with consistent trigger words and provide limited syntactic features. Titles lack context and vary in structure and organization. To the best of our knowledge, there is no publicly available up-to-date academic knowledgebase to guide the extraction task. Furthermore, it is hard to generate labeled domain-specific corpora to train supervised NER frameworks on academic text unlike general textual corpora. This makes our problem fundamentally challenging and interesting to solve. The key requirements of our technique are as follows:

- Independent of supervision via annotated academic text or human curated external resources.
- Flexible and generalizable to diverse academic domains.
- Independent of apriori parameters such as length of concept mentions, number of concepts corresponding to each aspect etc.

Unlike article text, titles lack contextual information and provide limited textual features rendering conventional NP-chunking [23] and dependency parsing [4] based extraction methods ineffective. Previous work in academic concept extraction [4, 20, 23] typically perform aspect or facet typing post extraction of concepts. Alternately, we propose to type phrases rather than individual concept mentions, and subsequently extract concepts from typed phrases. Phrases combine concept mentions such as tcp with additional specializing text e.g improving tcp performance, which provides greater clarity in aspect-typing the phrase as an application, rather than

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM’17, November 6–10, 2017, Singapore.
© 2017 ACM. ISBN 978-1-4503-4918-5/17/11... $15.00
DOI: https://doi.org/10.1145/3132847.3133023
We now define terminology used in this paper and formalize our aspects. Academic concepts require instance-specific aspect typing. Dirichlet Process could both be studied as a problem (Application) as well as proposed as a solution (Technique).

**Relation Phrase**: A relation phrase denotes a unary or binary relation which associates multiple phrases within a title. Extracting textual relations and applying them to entity typing has been studied in previous work [9, 12]. We use the left and right relation phrases connecting a phrase, as features to perform aspect typing of the phrase.

**Phrases**: Phrases are contiguous chunks of words separated by relation phrases within a title. Phrases could potentially contain concept mentions and other specializing or modifying words.

**Modifier**: Modifiers are specializing terms or subphrases that appear in conjunction with concept mentions within phrases. For instance, *Time based* is a modifier for the concept *language model* in the phrase *Time based language model* as illustrated in Fig. 4.

**Definition 2.1. Problem Definition**: Given an input collection $\mathcal{D}$ of titles of articles, a finite set of aspects $\mathcal{A}$, our goal is to:
1) Extract and partition the set of phrases $P$ from $\mathcal{D}$ into $|\mathcal{A}|$ subsets. Each aspect of interest in $\mathcal{A}$ is mapped to one subset of the partition by a mapping $M$.
2) Extract concept mentions and modifiers from each of the $|P|$ aspect-typed phrases. Concept mentions are ascribed the aspect type of the phrase in which they appear.

We achieve the above two goals in two phases of our algorithm, the first phase being Phrase Typing and the second, Fine Grained Concept Extraction. The output of our algorithm is a set of typed concept mentions $C_d \forall d \in \mathcal{D}$ and their corresponding modifier subphrases.

### 3 PHRASE TYPING

In this section, we describe our unsupervised approach to extract and aspect-type scientific phrases.

#### 3.1 Phrase segmentation

Input scientific titles are segmented into a set of phrases, and their connecting relation phrases that separate them within the title. We apply part-of-speech tag patterns similar to [20] to identify relation phrases. Additionally, we note here that not every relation phrase is appropriate for segmenting a title. Pointwise Mutual
Algorithm 1: PhraseType algorithm

1: Draw overall aspect distribution in the corpus, \( \theta \sim \text{Dir}(\alpha) \)
2: for each aspect \( a \) do
3: \hspace{1em} Choose unigram distribution \( \phi_w^a \sim \text{Dir}(\beta_w) \)
4: \hspace{1em} Choose significant phrase distribution \( \phi^a \sim \text{Dir}(\beta_w) \)
5: \hspace{1em} Choose left relation phrase distribution \( \phi^a_{l} \sim \text{Dir}(\beta_l) \)
6: \hspace{1em} Choose right relation phrase distribution \( \phi^a_{r} \sim \text{Dir}(\beta_r) \)
7: for each phrase \( p \) do
8: \hspace{1em} Choose aspect \( a \sim \text{Mult}(\theta) \)
9: \hspace{1em} for each token \( i = 1...|p_w| \) do
10: \hspace{2em} draw \( w_i \sim \text{Mult}(\phi_w^a) \)
11: \hspace{1em} for each significant phrase \( j = 1...|p_{sp}| \) do
12: \hspace{2em} draw \( sp_j \sim \text{Mult}(\phi^a_{sp}) \)
13: \hspace{1em} if \( p_l \) exists then draw \( p_l \sim \phi^a_{l} \)
14: \hspace{1em} if \( p_r \) exists then draw \( p_r \sim \phi^a_{r} \)

3.2 PhraseType

Relation phrases play consistent roles in paper titles and provide strong cues on the aspect role of a candidate phrase. A relation phrase such as by applying is likely to link a problem phrase to a solution. However not all titles contain informative relation phrases. Furthermore, we find that 19% of all titles in our corpus contain no relation phrases. Thus, it is necessary to build a model that combines relation phrases with textual features and learns consistent associations of aspects and text. To this end, we propose a flexible probabilistic generative model PhraseType which models the generation of phrases jointly over available evidence.

Each phrase is assumed to be drawn from a single aspect and the corresponding textual features and connecting relation phrases are obtained by sampling from the respective aspect distributions. Aspects are described by their distributions over left and right relation phrases and textual features including unigrams (filtered to remove stop words and words with very low corpus IDF) and significant multi-word phrases. Significant phrases are defined in a manner similar to [3] and extracted at the corpus level. Left and right relation phrases are modeled as separate features to factor associations of the phrase with adjacent phrases.

For each phrase \( p \) present in the corpus, we choose \( p_w \) to denote the set of tokens in \( p \), \( p_{sp} \) the set of significant phrases in \( p \), and \( p_l \), \( p_r \) the left and right relation phrases of \( p \) respectively. The generative process for a phrase is described in Alg 1 and the corresponding graphical representation in Fig 2 (For the sake of brevity we merge \( \phi_{sp} \) and \( \phi_w \) in Fig 2).

### 3.3 DomainPhraseType

Most academic domains significantly differ in the scope and content of published work. Modeling aspects at a domain-specific granularity is likely to better disambiguate phrases into appropriate aspects. A simplification could be to use venues directly as domains, however resulting in sparsity issues and not capturing interdisciplinary

![Figure 2: Graphical model for PhraseType](image)

Figure 2: Graphical model for PhraseType

work well. Most popular venues also contain publications on several themes and diverse tracks. We thus integrate venues and textual features in a common latent framework. This enables us to capture cross-venue similarities and yet provides room to discover diverse intra-venue publications and place them in appropriate domains. To this end, we present DomainPhraseType which extends PhraseType by factoring domains in the phrase generation process.

To distinguish aspects at a domain-specific granularity, it is necessary to learn textual features specific to a (domain, aspect) pair. Relation phrases however are domain-independent and play a consistent role with respect to different aspects. Additionally, venues often encompass several themes and tracks, although they are fairly indicative of the broad domain of study. Thus, we model domains as simultaneous distributions over aspect-specific textual features, as well as venues. Unlike PhraseType, textual features of phrases are now drawn from domain-specific aspect distributions, enabling independent variations in content across domains. The resulting generative process is summarized in Alg 2 and the corresponding graphical model in Fig 3. Parameter \(|D|\) describes the number of domains in the corpus \( D \).

3.4 Post-Inference Typing

In the PhraseType model, we compute the posterior distribution over aspects for each phrase as,

\[
P(a | p) \propto P(a) P(p_l | a) P(p_r | a) \prod_{i=1}^{p_w} P(w_i | a) \prod_{j=1}^{p_{sp}} P(sp_j | a)
\]

and assign it to the most likely aspect. Analogously, in DomainPhraseType, we compute the likelihood of (domain, aspect) pairs for each phrase,

\[
P(d, a | p) \propto P(d) P(p_{w_c} | d) P(a) P(p_l | a) P(p_r | a) \prod_{i=1}^{p_w} P(w_i | d, a) \prod_{j=1}^{p_{sp}} P(sp_j | d, a)
\]
Algorithm 2 DomainPhraseType algorithm

1. Draw overall aspect and domain distributions for the corpus, \( \theta^A \sim \text{Mult}(\alpha^A) \) and \( \theta^D \sim \text{Mult}(\alpha^D) \).

2. for each aspect \( a \) do

3. Choose left relation phrase distribution, \( \phi_l^a \sim \beta_l \)

4. Choose right relation phrase distribution, \( \phi_r^a \sim \beta_r \)

5. for each domain \( d \) do

6. Draw domain-specific venue distribution \( \phi_v^d \sim \text{Dir}(\beta_v) \)

7. for each aspect \( a \) do

8. Choose unigram distribution \( \phi_u^a \sim \text{Dir}(\beta_u) \).

9. Choose significant phrase distribution \( \phi_s^a \sim \text{Dir}(\beta_s) \).

10. for each phrase \( p \) do

11. Choose aspect \( a \sim \text{Mult}(\theta^A) \) and domain \( d \sim \text{Mult}(\theta^D) \)

12. for each token \( i = 1 \ldots |p| \) do

13. Draw \( w_i \sim \text{Mult}(\phi_w^d,a) \)

14. for each significant phrase \( j = 1 \ldots |p| \) do,

15. Draw \( sp_j \sim \text{Mult}(\phi_s^d,a) \)

16. Draw venue \( v \sim \phi_v^d \).

17. if \( p_l \) exists then draw \( p_l \sim \phi_l^a \).

18. if \( p_r \) exists then draw \( p_r \sim \phi_r^a \).

and assign the most likely pair. Phrases with consistently low posteriors across all pairs are discarded.

Additionally we must now map the aspects \( a \in [1, |\mathcal{A}|] \) inferred by our model to the aspects of interest, i.e. \( \mathcal{A} \) by defining mapping \( \mathcal{M} \) from \( \mathcal{A} \) to \( [1, |\mathcal{A}|] \). Note that there are \( |\mathcal{A}|! \) possible ways to do this, however \( |\mathcal{A}| ! \) is a small number in practice. Although our model provides the flexibility to learn any number of aspects, we find that most concept mentions in our datasets are sufficiently differentiated into Techniques and Applications by setting parameter \( |\mathcal{A}| \) to 2 in both our models. In other domains such as medical literature, it might be appropriate to learn more than two aspects to partition phrases in medical text. Let 1 and 2 denote the aspects inferred, and \( \mathcal{A} = [\text{Technique(T)}, \text{Application(A)}] \). We use the distributions \( \phi_l \) and \( \phi_r \) of the inferred aspects to set mapping \( \mathcal{M} \) either to \( \mathcal{M}(T, A) = (1, 2) \) or \( \mathcal{M}(T, A) = (2, 1) \). Strongly indicative relation phrases such as by using and by applying are very likely to appear at the left of the Technique phrase of a title, and at the right of the Application phrase. Given a set of indicative relation phrases \( RP \), which are likely to appear as left relation phrases of Technique phrases, and right relation phrases of Application phrases, \( M \) is chosen to maximize the following objective:

\[
\mathcal{M} = \arg \max_{r_p \in RP} (\langle \phi_l(r_p) \rangle_{\mathcal{M}(T)} + [\phi_r(r_p)]_{\mathcal{M}(A)})
\]

3.5 Temporal Dependencies

Modeling the temporal evolution of domains is necessary to capture variations that arise over time, in the set of techniques and applications studied by articles published at various venues. To this end, we learn multiple models corresponding to varying time intervals, and explicitly account for expected contiguity in near time-slices. Our objectives with regard to temporal variations are two fold:

- Sufficient flexibility to describe varying statistical information over different time periods.
- Smooth evolution of statistical features in a given domain over time.

We therefore extend the above models in the time dimension. Our dataset is partitioned into multiple time-slices with roughly the same number of articles. Both models follow the generative processes described above on all phrases in the first time-slice. For subsequent slices the target phrases are modeled in a similar generative manner, however text and venue distributions \( \phi_w^d,a \) and \( \phi_v^d \) are described by a weighted mixture of the corresponding distributions learned in the previous time-slice, in addition to the prior. This enables us to maintain a connection between domains and aspects learned in different time-slices while also providing flexibility to account for new applications and techniques. Thus \( T \geq 2 \):

\[
\begin{align*}
\phi_w^d,a_{T-\omega} &\sim \omega \phi_w^d,a_{T-1} + (1-\omega) \text{Dir}(\beta_w) \\
\phi_v^d_{T-\omega} &\sim \omega \phi_v^d_{T-1} + (1-\omega) \text{Dir}(\beta_v) \\
\phi_u^d_{T-\omega} &\sim \omega \phi_u^d_{T-1} + (1-\omega) \text{Dir}(\beta_u)
\end{align*}
\]

4 FINE GRAINED CONCEPT EXTRACTION

Academic phrases are most often composed of concepts and modifying subphrases in arbitrary orderings. Concept mentions appear as contiguous units within phrases and are trailed or preceded by modifiers. Thus our concept extraction problem can be viewed as shallow parsing or chunking [1] of phrases. Unlike grammatical sentences or paragraphs, phrases lack syntactic structure, and the vast majority of them are composed of noun phrases or proper nouns and adjectives. Thus classical chunking models are likely to perform poorly on these phrases.

Unlike generic text fragments, our phrases are most often associated with key atomic concepts which do not display variation in word ordering and always appear as contiguous units across the corpus. For instance, concepts such as hierarchical clustering or peer to peer network always appear as a single chunk, and are preceded

Figure 3: Graphical model for DomainPhraseType
and followed by modifiers e.g. Incremental hierarchical clustering or Analysis of peer to peer network. This property motivates us to parse phrases with simple rule-based grammars, by statistically discovering concepts in the dataset.

Probabilistic Context-Free Grammars (PCFGs) are a statistical extension of Context Free Grammars [2] that are parametrized with probabilities over production rules, which leads to probability distributions over the possible parses of a phrase. However the independence assumptions render them incapable of learning parses dynamically. Their non-parametric extension, adaptor grammars [7], can cache parses to learn derivations of phrases in a data-driven manner. Furthermore they are completely unsupervised, which negates the need for any human effort in annotating concepts or training supervised NER frameworks. In the following section, we briefly describe PCFGs and adaptor grammars, and their application to extracting concept mentions and modifiers from phrases.

### 4.1 Probabilistic Context-free Grammars

A PCFG $G$ is defined as a quintuple $(N, W, R, S, \theta)$. Given a finite set of terminals $W$, nonterminals $N$ and start symbol $S$, $G$ is given by a set of probabilistic grammar rules $(R, \theta)$ where $R$ represents a set of grammar rules while $\theta$ is the set of probabilities associated with each rule. Let $R_A$ denote the set of all rules that have a nonterminal $A$ in the head position. Each grammar rule $A \rightarrow \beta$ is also called a production and is associated with a corresponding probability $\theta_{A \rightarrow \beta}$ which is the probability of expanding the nonterminal $A$ using the production $A \rightarrow \beta$. According to the definition of a PCFG, we have a normalization constraint for each non-terminal:

$$\sum_{A \rightarrow \beta} \theta_{A \rightarrow \beta} \in R_A = 1 \forall A \in N$$

The generation of a sentence belonging to the grammar starts from symbol $S$ and each non-terminal is recursively re-written into its derivations according to the probabilistic rules defined by $(R, \theta)$. The rule to be applied at each stage of derivation is chosen independently (of the existing derivation) based on the production probabilities. This results in a hierarchical derivation tree, starting from the start symbol and resulting in a sequence of terminals in the leaf nodes. The final sequence of terminals obtained from the parse tree is called the yield of the derivation tree. A detailed description of PCFGs can be found here [11].

### 4.2 Adaptor Grammars

PCFGs build derivation trees for each parse independently with a predefined probability on each rule ignoring the yields and structure of previously derived parse trees to decide on rule derivation. For instance, the derivation tree Concept → language model highlighted in Fig 4, cannot be learned by a PCFG since every phrase containing language model is parsed independently. Adaptor grammars address this by augmenting the probabilistic rules of a PCFG to capture dependencies among successive parses. They jointly model the context and the grammar rules in order to break the independence assumption of PCFGs by caching derivation trees corresponding to previous parses and dynamically expanding the set of derivations in a data-driven fashion.

Concept mentions such as language model are likely to appear in several parses and are hence cached by the grammar, which in turn ensures consistent parsing and extraction of the most significant concepts across the corpus. In addition, it has the advantage of being a non-parametric Bayesian model in contrast to PCFG which is parametrized by rule probabilities $\theta$. Adaptor Grammars (Pitman-Yor Grammars) dynamically learn meaningful parse trees for each adapted nonterminal from the data based on the Pitman-Yor process (PYP) [13]. Formally, Pitman-Yor Grammar $P \mathcal{Y} G$ is defined as,

- Finite set of terminals $W$, nonterminals $N$, rules $R$ and start symbol $S$.
- Dirichlet prior $\alpha_A$ for the production probabilities $\theta_A$ of each nonterminal $A \in N$, $\theta_A \sim Dir(\alpha_A)$.
- Set of non-recursive adaptors $C \subseteq N$ with PYP parameters $a_c, b_c$ for each adaptor $c \in C$.

The Chinese Restaurant Process (CRP) [6] provides a realization of PYP described by a scale parameter, $a$, discount factor $b$ and a base distribution $G_c$ for each adaptor $c \in C$. The CRP assumes that dishes are served on an unbounded set of tables, and each customer entering the restaurant decides to either be seated on a pre-occupied table, or a new one. The dishes served on the tables are drawn from the base distribution $G_c$. CRP sets up a rich get richer dynamics, i.e. new customers are more likely to occupy crowded tables. Assume that when the $N^h$ customer enters the restaurant, the previous $N-1$ customers labeled $[1, 2, ..., N-1]$ have been seated on $K$ tables ($K \leq N-1$), and the $i^{th}$ customer be seated on table $x_i \in [1, ..., K]$. The $N^h$ customer chooses to sit at $x_N$ with the following distribution (note that if he chooses an empty table, this is now the $K + 1^{st}$ table),

$$P(x_N | x_1, ..., x_{N-1}) \sim \frac{Kb + a}{N - 1 + a} \delta_{K+1} + \sum_{k=1}^{K} \frac{m_k - b}{N - 1 + a} \delta_k$$

where,

$$m_k = \#x_i, i \in [1, ..., N-1], x_i = k$$

where $\delta_{K+1}$ refers to the case when a new table is chosen. Thus the customer chooses an occupied table with a probability proportional to the number of occupants ($m_k$) and an unoccupied table proportional to the scale parameter $a$ and the discount factor $b$. It can be shown that all customers in CRP are mutually exchangeable and do not alter the distribution. Thus the probability distribution of any sequence of table assignments for customers depends only on the number of customers per table $n = \{n_1, ..., n_K\}$. This probability is
4.3 Inference

The objective of inference is to learn a distribution over derivation trees given a collection of phrases as input. Let $P$ be the collection of phrases and $T$ be the set of derivation trees used to derive $P$. The probability of $T$ is then given by,
\[
P(T | a, b) = \prod_{A \in N - C} P_{dir}(f_A(T) | a_A) \prod_{c \in C} P_{py}(n_c(T) | a_c, b_c)
\]

where $n_c(T)$ represents the frequency vector of all adapted rules for adaptor $c$ being observed in $T$ and $f_k(T)$ represents the frequency vector of all PCFG rules for non-terminal $A$ being observed in $T$. Here, $P_{py}(n | a, b)$ is as given in Eqn. 1, while the dirichlet posterior probability $P_{dir}(f | a)$ for a given nonterminal is given by,
\[
P_{dir}(f | a) = \frac{\Gamma(\sum_{k=1}^{K} a_k) \prod_{k=1}^{K} \Gamma(f_k + a_k)}{\Gamma(\sum_{k=1}^{K} f_k + a_k) \prod_{k=1}^{K} \Gamma(a_k)}
\]

where $K = |R_A|$ is the number of PCFG rules associated with $A$, and variables $f$ and $a$ are both vectors of size $K$. Given an observed string $x$, in order to compute the posterior distribution over its derivation trees, we need to normalize $P(T | a, b)$ over all derivation trees that yield $x$. Computing this distribution directly is intractable. We use a MCMC Metropolis-Hastings [7] sampler to perform inference. We refer readers to [7, 8] for a detailed description of MCMC methods for adaptor grammar inference.

4.4 Grammar Rules

The set of phrases $P$ is partitioned by aspect in PhraseType and by aspect as well as domain, in case of DomainPhraseType. This provides us the flexibility to parse phrases of each aspect (and domain) with a different grammar. Furthermore, parsing each partition separately enables adaptors to recognize discriminative and significant concept mentions specific to each subset which is one of our primary motivations for typing phrases prior to concept extraction. Although a single grammar suffices in the case of Techniques and Application aspects, aspect-specific grammars could also be defined when phrases significantly differ in organization or structure, within our framework.

Since phrases are obtained by segmenting titles on relation phrases, it is reasonable to assume that in most cases there is at-most one significant concept mention in a phrase. The set of productions of the adaptor grammar used are illustrated in Fig 5 (Adaptor), with Concept being the adapted non-terminal. We also experiment with a variation where both Concept and Mod are adapted (Adaptor:Mod). It appears intuitive to adapt both non-terminals since several modifiers are also statistically significant, such as high dimensional, Analyzing, low rank etc. However, our experimental results appear to indicate that adapting Concept alone performs better. Owing to the structure of the grammar, a competition is set up between Concepts and Mods when both are adapted. This causes a few instances of phrases such as low rank matrix representation to be partitioned incorrectly between Mod and Concept causing a mild degradation in performance. When Concept alone is adapted, the most significant subphrase matrix representation is extracted as the Concept as expected.

5 EXPERIMENTS

We evaluate the effectiveness and scalability of our concept extraction framework by conducting experiments on two real-world datasets : DBLP and ACL [16].

5.1 Experimental setup

79 top conferences were chosen in the DBLP dataset from diverse domains including NLP & Information Retrieval (IR), Artificial Intelligence and Machine Learning (ML), Databases and Data Mining (DM), Theory and Algorithms (ALG), Compilers and Programming Languages (PL) and Operating Systems & Computer Networks (NW). The top 50 venues by number of publications were chosen for the ACL dataset. We focus on two primary evaluation tasks.

Quality of concepts:
We evaluate the quality of concept mentions identified by each method, without considering the aspect. A set of multi-domain gold standard concepts were chosen from the ACL and DBLP datasets. A random sample of 2,381 documents (for DBLP) and 253 documents (for ACL) containing the chosen gold standard concepts were chosen for evaluation.

Identification of aspect-typed concept mentions:
We evaluate the final result set of aspect-typed concept mentions identified by each method on both domain-specific as well as multi-domain corpora. Methods are given credit if both the concept mention as well as the aspect assigned to it are correct. To perform domain-specific analysis, we manually partition the set of titles in the DBLP dataset into 6 categories based on the venues, and use the unpartitioned DBLP and ACL datasets directly for multi-domain experiments.

A subset of titles in each dataset were annotated with typed concept mentions appearing in their text. Each concept mention
was identified and typed to the most appropriate aspect among Technique and Application independently by a pair of experts. The inter-annotator agreement (kappa-value) was found to be 0.86 on DBLP and 0.93 on ACL and the titles where the annotators agreed were chosen for evaluation. Table 1 summarizes the details of corpus and gold standard annotations. Our gold-standard annotations are publicly available online.

Evaluation Metrics:
For concept quality evaluation, we compute the F1 score with Precision and Recall. Precision is computed as the ratio of correctly identified concept mentions to the total number of identified mentions. Recall is defined as the ratio of correctly identified concept mentions to the total number of mentions of gold standard concepts in the chosen subset of documents.

For identification of typed concept mentions, precision is defined as the ratio of correctly identified and typed concept mentions to the total number of identified mentions. Recall is defined as the ratio of correctly identified and typed concept mentions to the total number of typed concept mentions chosen by the experts.

Baselines:
To evaluate concept quality, we compare against two mention extraction techniques in literature - Shallow parsing and Phrase Segmentation. Specifically, we compare against: 1) Noun Phrase (NP) chunking and 2) SegPhrase [10]. To evaluate identification of aspect-typed concept mentions, we compare our algorithms with multiple strong baselines:

- **Bootstrapping + NP chunking** [23]: This is a bootstrapping based concept extraction approach and is currently the state-of-the-art technique for concept extraction in scientific literature.
- **Bootstrapping + SegPhrase**: We use a phrase-segmentation algorithm SegPhrase [10] to generate candidate concept mentions and apply the above bootstrapping algorithm to extract typed concepts.
- **PhraseType + PCFG**: We use PhraseType combined with a PCFG grammar to extract aspect-typed concepts.
- **PhraseType + Adaptor**: This uses our PhraseType model to extract aspect-typed phrases and performs concept extraction using the Adaptor grammar defined in Fig 5 with Concept being adapted.
- **DomainPhraseType + Adaptor**: This uses DomainPhraseType to extract aspect-typed phrases and performs concept extraction independently for each domain using the productions defined in Fig 5 with Concept being adapted.
- **DomainPhraseType + Adaptor:Mod**: This uses DomainPhraseType as above and performs concept extraction using the productions defined in Fig 5 while adapting both Mod and Concept non-terminals.

For the bootstrapping algorithms, we use a development set of 20 titles in each dataset and set the parameters \((k, n, t)\) to \((200, 200, 2)\) as recommended in [23]. For PhraseType, we set parameters \(\alpha = 50/|A|\) and \(\beta = \beta_l = \beta_r = 0.01\), while for DomainPhraseType, we set \(\alpha_A = 50/|A|\), \(\alpha_D = 50/|D|\) and \(\beta_w = \beta_l = \beta_r = \beta_v = 0.01\) and perform inferencing with collapsed gibbs sampling. Temporal parameter \(\phi\) was set to 0.5. In our experiments, we run mcmc samplers for 1000 iterations. For DomainPhraseType, we varied the number of domains for each dataset and found that \(|D| = 10\) in DBLP and \(|D| = 5\) in ACL result in the best F1-scores (Fig. 6(b)). Discount and scale parameters of adaptors \((a, b)\) were set to \((0.5, 0.5)\) in both Adaptor and Adaptor:Mod and dirichlet prior \(\alpha_A\) is set to 0.01.

### Table 1: Dataset and Gold Standard statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DBLP</th>
<th>ACL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Titles</td>
<td>188974</td>
<td>14840</td>
</tr>
<tr>
<td>Venues</td>
<td>79</td>
<td>50</td>
</tr>
<tr>
<td>Gold Standard titles</td>
<td>740</td>
<td>100</td>
</tr>
<tr>
<td>Gold Standard Technique</td>
<td>630</td>
<td>96</td>
</tr>
<tr>
<td>Gold Standard Application</td>
<td>783</td>
<td>108</td>
</tr>
</tbody>
</table>

### Table 2: Concept quality performance comparison with baselines on DBLP and ACL

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>DBLP</th>
<th>ACL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP chunking</td>
<td>Prec</td>
<td>0.483</td>
<td>0.652</td>
</tr>
<tr>
<td></td>
<td>Rec</td>
<td>0.292</td>
<td>0.376</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>0.364</td>
<td>0.477</td>
</tr>
<tr>
<td>SegPhrase</td>
<td>Prec</td>
<td>0.509</td>
<td>0.740</td>
</tr>
<tr>
<td></td>
<td>Rec</td>
<td>0.279</td>
<td>0.451</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>0.360</td>
<td>0.573</td>
</tr>
<tr>
<td>PhraseType + Adaptor</td>
<td>Prec</td>
<td>0.699</td>
<td>0.623</td>
</tr>
<tr>
<td></td>
<td>Rec</td>
<td>0.739</td>
<td>0.644</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>0.718</td>
<td>0.633</td>
</tr>
<tr>
<td>DomainPhraseType + Adaptor:Mod</td>
<td>Prec</td>
<td>0.806</td>
<td>0.732</td>
</tr>
<tr>
<td></td>
<td>Rec</td>
<td>0.731</td>
<td>0.694</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>0.767</td>
<td>0.713</td>
</tr>
</tbody>
</table>

5.2 Experimental Results

**Quality of concepts**: As depicted in Table 2, the concept extraction techniques based on adaptor grammars indicate a significant performance gain over other baselines on both datasets. Adaptor grammars exploit corpus-level statistics to accurately identify the key concept mentions in each phrase which leads to better quality concept mentions in comparison to shallow parsing and phrase segmentation. Amongst the baselines, we find SegPhrase to have a high precision since it extracts only high quality phrases from the titles while all of them suffer from poor recall due to their inability to extract fine-grained concept mentions accurately.

We find PhraseType + Adaptor to outperform DomainPhraseType + Adaptor by a small margin. PhraseType + Adaptor is able to extract concepts of higher quality since it is learned on the entire corpus while DomainPhrase + Adaptor performs concept extraction specific to each domain and could face sparsity in some domains, however this is offset by improved aspect typing by DomainPhraseType + Adaptor in the identification of typed concept mentions.

**Identification of aspect-typed concept mentions**: For aspect-typed concept mention identification, we first evaluate the performance of PhraseType + Adaptor against the baselines on domain-specific subsets of DBLP (Table 3). We then evaluate all techniques including DomainPhraseType + Adaptor/Adaptor:Mod on the complete multi-domain ACL and DBLP datasets (Table 4). We find DomainPhraseType based methods to outperform PhraseType owing to improved aspect typing at the domain granularity.

---

1. [https://sites.google.com/site/conceptextraction2/](https://sites.google.com/site/conceptextraction2/)
Table 3: DBLP: Domain-specific results (Precision, Recall and F1 scores) - comparing PhraseType with baselines

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>IR</th>
<th>ML</th>
<th>DM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Prec</td>
<td>Rec</td>
<td>F1-Score</td>
</tr>
<tr>
<td>DBLP</td>
<td>Bootstrapping + NP</td>
<td>0.437</td>
<td>0.325</td>
<td>0.373</td>
</tr>
<tr>
<td></td>
<td>Bootstrapping + Segphrase</td>
<td>0.717</td>
<td>0.497</td>
<td>0.587</td>
</tr>
<tr>
<td></td>
<td>PhraseType + PCFG</td>
<td>0.444</td>
<td>0.487</td>
<td>0.465</td>
</tr>
<tr>
<td></td>
<td>PhraseType + Adaptor:Mod</td>
<td>0.599</td>
<td>0.669</td>
<td>0.632</td>
</tr>
<tr>
<td></td>
<td>PhraseType + Adaptor</td>
<td>0.712</td>
<td>0.793</td>
<td>0.750</td>
</tr>
<tr>
<td>ACL</td>
<td>Bootstrapping + NP</td>
<td>0.548</td>
<td>0.398</td>
<td>0.461</td>
</tr>
<tr>
<td></td>
<td>Bootstrapping + Segphrase</td>
<td>0.617</td>
<td>0.425</td>
<td>0.503</td>
</tr>
<tr>
<td></td>
<td>PhraseType + PCFG</td>
<td>0.478</td>
<td>0.478</td>
<td>0.478</td>
</tr>
<tr>
<td></td>
<td>PhraseType + Adaptor:Mod</td>
<td>0.576</td>
<td>0.569</td>
<td>0.572</td>
</tr>
<tr>
<td></td>
<td>PhraseType + Adaptor</td>
<td>0.604</td>
<td>0.607</td>
<td>0.605</td>
</tr>
</tbody>
</table>

Table 4: DBLP, ACL: Precision, Recall and F1 scores - Performance comparisons with baselines on individual aspects

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Application</th>
<th>Technique</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Prec</td>
<td>Rec</td>
<td>F1-Score</td>
</tr>
<tr>
<td>DBLP</td>
<td>Bootstrapping + NP</td>
<td>0.330</td>
<td>0.323</td>
<td>0.326</td>
</tr>
<tr>
<td></td>
<td>Bootstrapping + Segphrase</td>
<td>0.418</td>
<td>0.432</td>
<td>0.425</td>
</tr>
<tr>
<td></td>
<td>PhraseType + PCFG</td>
<td>0.369</td>
<td>0.381</td>
<td>0.375</td>
</tr>
<tr>
<td></td>
<td>DomainPhraseType + PCFG</td>
<td>0.604</td>
<td>0.628</td>
<td>0.616</td>
</tr>
<tr>
<td></td>
<td>DomainPhraseType + Adaptor</td>
<td>0.603</td>
<td>0.618</td>
<td>0.610</td>
</tr>
<tr>
<td></td>
<td>DomainPhraseType + Adaptor:Mod</td>
<td>0.657</td>
<td>0.692</td>
<td>0.674</td>
</tr>
<tr>
<td>ACL</td>
<td>Bootstrapping + NP</td>
<td>0.283</td>
<td>0.265</td>
<td>0.274</td>
</tr>
<tr>
<td></td>
<td>Bootstrapping + Segphrase</td>
<td>0.655</td>
<td>0.582</td>
<td>0.616</td>
</tr>
<tr>
<td></td>
<td>PhraseType + PCFG</td>
<td>0.326</td>
<td>0.316</td>
<td>0.321</td>
</tr>
<tr>
<td></td>
<td>DomainPhraseType + PCFG</td>
<td>0.645</td>
<td>0.612</td>
<td>0.628</td>
</tr>
<tr>
<td></td>
<td>DomainPhraseType + Adaptor:Mod</td>
<td>0.412</td>
<td>0.408</td>
<td>0.410</td>
</tr>
<tr>
<td></td>
<td>DomainPhraseType + Adaptor</td>
<td>0.730</td>
<td>0.745</td>
<td>0.737</td>
</tr>
</tbody>
</table>

Table 5: Modifiers for a few sample concepts

<table>
<thead>
<tr>
<th>Concept</th>
<th>Modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approximation algorithm</td>
<td>Improved, Constant-Factor, Polynomial-Time, Stochastic, Distributed, Adaptive</td>
</tr>
<tr>
<td>Decision tree</td>
<td>Induction, Learning, Classifier, Algorithm, Cost-Sensitive, Pruning, Construction, Boosted</td>
</tr>
<tr>
<td>Wireless network</td>
<td>Multi-Hop, Heterogeneous, Ad-Hoc, Mobile, Multi-Channel, Large, Cooperative</td>
</tr>
<tr>
<td>Topic model</td>
<td>Probabilistic, Supervised, Latent, Approach, Hierarchical, LDA, Biterm, Statistical</td>
</tr>
<tr>
<td>Neural network</td>
<td>Recurrent, Convolutional, Deep, Approach, Classifier, Architecture</td>
</tr>
<tr>
<td>Sentiment analysis</td>
<td>Aspect-Based, Cross-Linguual, Sentence-Level, In-Twitter, Unsupervised</td>
</tr>
<tr>
<td>Image classification</td>
<td>Large-scale, Fine-grained, Hyperspectral, Multi-Label, Simultaneous, Supervised</td>
</tr>
</tbody>
</table>

Effect of corpus size on performance: We vary the size of the DBLP dataset by randomly sampling a subset of the corpus in addition to the gold-standard annotated titles and measure the performance of different techniques (Fig 6(a)). We observe significant performance drop when the size of the corpus is reduced to ≤ 20% of all titles, primarily due to reduced representation of sparse domains. Performance appears to be stable post 30%.

Effect of number of domains: To observe the effect of number of domains on performance, we varied |D| from 1 to 20 in the DomainPhraseType model for the DBLP and ACL datasets as in Fig 6(b). Final results are reported based on the optimal number of domains, 10 for DBLP and 5 for ACL.

Runtime analysis: Our experiments were performed on an x64 machine with Intel(R) Xeon(R) CPU E5-345 (2.33GHz) and 16 GB of memory. All models were implemented in C++. Our runtime was found to vary linearly with the corpus size (Fig 6(c)).

5.3 Case Studies

Top modifiers for sample concepts: We extract the modifiers obtained by DomainPhraseType + Adaptor for a few sample concepts and depict the top modifiers (ranked by their popularity) in Table 5.
plied to the abstracts of scientific articles, assuming the presence of supervised setting where Bootstrapping algorithms [4, 23] are applied. Candidate mentions are typed to each domain. An interesting observation is the ability of our framework to distinguish between fine-grained domains such as IR and NLP and identify the most relevant concepts for each domain accurately.

6 RELATED WORK

The objective of our work is the automatic typing and extraction of concept mentions in short text such as paper titles, into aspects such as Technique and Application. Unlike typed entities in a traditional Named Entity Recognition (NER) setting such as people, organizations, places etc., academic concepts are not notable entity names that can be referenced from a knowledgebase or external resource. They exhibit variability in surface form and usage and evolve over time. Indicative features such as trigger words (Mr., Mrs. etc), grammar properties and predefined patterns are inconsistent or absent in most academic titles. Furthermore, NER techniques rely on rich contextual information and semantic structures of text. Paper titles, on the other hand, are structured to be succinct, and lack context words.

The problem of semantic class induction [18, 22] is related to typing of concept mentions since aspects are analogous to semantic classes. [24] studies the extraction of generalized names in the medical domain through a bootstrapping approach, however academic concepts are more ambiguous and hence harder to type. Many of them correspond to both Technique and Application aspects in different mentions, and hence must be typed in an instance specific manner rather than globally. To the best of our knowledge there has been very limited work in extraction of typed concept mentions from scientific titles or abstracts.

Phrase mining techniques such as [3] and [10] study the extraction of significant phrases from large corpora, however they do not factor aspects or typing of phrases in the extraction process. We briefly summarize past approaches for academic concept extraction from the abstracts of articles. We also survey techniques that extract concept mentions within the full text of the article, which is not our primary focus.

Concept typing has been studied in earlier work in the weakly supervised setting where Bootstrapping algorithms [4, 23] are applied to the abstracts of scientific articles, assuming the presence of a seed list of high-quality concept mention instances for each aspect of interest. [4] uses dependency parses of sentences to extract candidate mentions and applies a bootstrapping algorithm to extract three types of aspects - focus, technique, and application domain. [23] uses noun-phrase chunking to extract concept mentions and local textual features to annotate concept mentions iteratively. Our experiments indicate that their performance is dependent on seeding domain-specific concepts. Furthermore, noun-phrase chunkers are dependent on annotated academic corpora for training. [20] extracts facet concept mentions in the article text by exploiting several sources of information including the structure of the paper, sectional information, citation data and other textual features. However it is hard to quantify the importance of each extracted facet or entity mention to the overall contribution or purpose of the scientific article.

Topic models have also been recently used to study the popularity of research communities and evolution of topics over time in scientific literature [21]. However, topic models that rely on statistical distributions over unigrams [14, 25, 27] do not produce sufficiently tight concept clusters in academic text. Citation based methods have also been used to analyze research trends [15], however their key focus is understanding specific citations rather than extracting the associated concepts. Attribute mining [5] combines entities and aspects (attributes) based on an underlying aspect hierarchy. Our work however identifies aspect-specific concept mentions at an instance level. [26] proposes an unsupervised approach based on pitman-yor grammars [7] to extract brand and product entities from shopping queries. However, brand and product roles are not interchangeable (a brand can never be a product) unlike academic concepts. Furthermore, most shopping queries are structured to place brands before product. Paper titles however are not uniformly ordered and thus need to be normalized by aspect typing their constituent phrases prior to concept extraction.

7 CONCLUSION

In this paper, we address the problem of concept extraction and categorization in scientific literature. We propose an unsupervised, domain-independent two-step algorithm to type and extract key concept mentions into aspects of interest. PhraseType and DomainPhraseType leverage textual features and relation phrases to type phrases. This enables us to extract aspect and domain specific concepts in a data-driven manner with adaptor grammars. While our
focus here has been to apply our algorithm on scientific titles to discover techniques and application aspects, there is potential to apply a similar two-step process in other domains such as medical text to discover aspects such as drugs, diseases, and symptoms. It is also possible to extend the models to sentences in full text documents while exploiting grammatical and syntactic structures. Our broader goal is to eliminate the need for human effort and supervision in domain-specific tasks such as ours.

8 ACKNOWLEDGMENTS

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 16-18481 and NSF IIS 17-04532, and grant IU54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative.

REFERENCES

[3] Ahmed El-Kishky, Yanglei Song, Chi Wang, Clare R Voss, and Jiawei Han. 2014. Domain Phrase Type in the DBLP dataset (D=10).