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 studying 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 atleast 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.

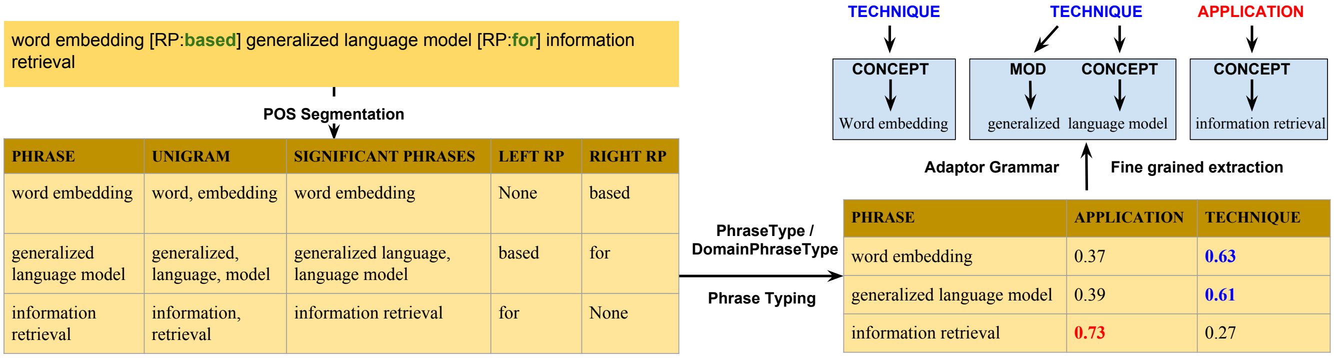
Our problem fundamentally differs from classic *Named Entity Recognition* techniques which focus on natural language text [17] and web resources via distant supervision [19]. Entity phrases corresponding to predefined categories such as person, organization, location etc are detected using trigger words (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

*Equal contribution

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



PHRASE	UNIGRAM	SIGNIFICANT PHRASES	LEFT RP	RIGHT RP
word embedding	word, embedding	word embedding	None	based
generalized language model	generalized, language, model	generalized language, language model	based	for
information retrieval	information, retrieval	information retrieval	for	None

PHRASE	APPLICATION	TECHNIQUE
word embedding	0.37	**0.63**
generalized language model	0.39	**0.61**
information retrieval	**0.73**	0.27

Figure 1: Pipeline of our concept extraction framework

the *tcp* concept mention. Phrases are structured with connecting relation phrases which can provide insights to their aspect roles, in conjunction with their textual content. Furthermore, aspect typing prior concept extraction provides us the flexibility to impose and learn aspect-specific concept extraction rules.

We thus propose a novel two-step framework that satisfies the above requirements. Our first contribution is an aspect-based generative model *PhraseType* to type phrases by learning representative textual features and the associated relation phrase structure. We also propose a domain-aware extension of our model *DomainPhraseType* by integrating domain identification and aspect inference in a common latent framework. Our second contribution is a data-driven non-parametric rule-based approach to perform fine-grained extraction of concept mentions from aspect-typed phrases, based on *adaptor grammars* [7]. We propose simple grammar rules to parse typed phrases and identify the key concept mentions accurately. The availability of tags from the previous step enables our grammars to learn aspect-specific parses of phrases.

To the best of our knowledge, ours is the first algorithm that can extract and type concept mentions from academic literature in an unsupervised setting. Our experimental results on over 200,000 multi-domain scientific titles from DBLP and ACL datasets show significant improvements over existing concept extraction techniques in both, typing as well as the quality of extracted concept mentions. We also present qualitative results to establish the utility of extracted concepts and domains.

2 PROBLEM DEFINITION

We now define terminology used in this paper and formalize our problem.

Concept: A concept is a single word or multi-word subphrase (we refer to it as a subphrase to distinguish it from phrases) that represents an academic entity or idea which is of interest to users (i.e it has a meaning and is significant in the corpus), similar to the definitions in [23] and [20]. Concepts are not unique in identity and multiple concepts could refer to the same underlying entity (e.g *DP* and *Dirichlet Process*).

Concept Mention: A concept mention is a specific occurrence or instance of a concept.

Aspects: Users search, read and explore scientific articles via attributes such as techniques, applications etc, which we refer to as 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 mention** *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 apsect of interest in $\mathcal{A}$ is mapped to one subset of the partition by a mapping $\mathcal{M}$.*
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

Information(PMI) measure can be applied to the preceding and following words to decide whether to split on a relation phrase or not. This ensures that coherent phrases such as *precision and recall* are not split.

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 level 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 ϕ_{sp} and ϕ_w in Fig 2).

Algorithm 1 PhraseType algorithm

```
Draw overall aspect distribution in the corpus, $\theta \sim \text{Dir}(\alpha)$
for each aspect $a$ do
    Choose unigram distribution $\phi_w^a \sim Dir(\beta_w)$
    Choose significant phrase distribution $\phi_{sp}^a \sim \text{Dir}(\beta_w)$
    Choose left relation phrase distribution $\phi_l^a \sim \text{Dir}(\beta_l)$
    Choose right relation phrase distribution $\phi_r^a \sim \text{Dir}(\beta_r)$
for each phrase $p$ do
    Choose aspect $a \sim \text{Mult}(\theta)$
    for each token $i = 1...|p_w|$ do
        draw $w_i \sim \text{Mult}(\phi_w^a)$
    for each significant phrase $j = 1...|p_{sp}|$ do
        draw $sp_j \sim \text{Mult}(\phi_{sp}^a)$
    if $p_l$ exists then draw $p_l \sim \phi_l^a$
    if $p_r$ exists then draw $p_r \sim \phi_r^a$
```

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 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.

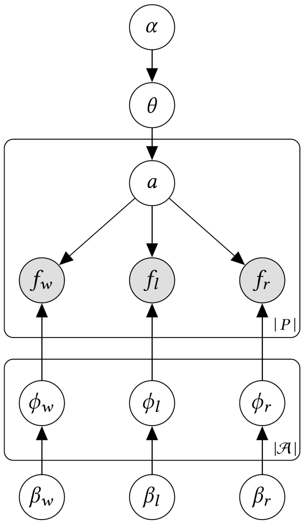


Figure 2: Graphical model for PhraseType

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 $\mathcal{D}$.

3.4 Post-Inference Typing

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

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

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

$$P(d, a \mid p) \propto P(d)\ P(p_v \mid d)\ P(a)\ P(p_l \mid a)\ P(p_r \mid a) \times \prod_{i=1}^{|p_w|} P(w_i \mid d, a) \prod_{j=1}^{|p_{sp}|} P(sp_j \mid d, a)$$

Algorithm 2 DomainPhraseType algorithm

```
Draw overall aspect and domain distributions for the corpus, $\theta^A \sim \text{Mult}(\alpha^A)$ and $\theta^D \sim \text{Mult}(\alpha^D)$
for each aspect $a$ do
    Choose left relation phrase distribution, $\phi_l^a \sim \beta_l$
    Choose right relation phrase distribution, $\phi_r^a \sim \beta_r$
for each domain $d$ do
    Draw domain-specific venue distribution $\phi_v^d \sim \text{Dir}(\beta_v)$
    for each aspect $a$ do
        Choose unigram distribution $\phi_w^{d,a} \sim \text{Dir}(\beta_w)$.
        Choose significant phrase distribution $\phi_{sp}^{d,a} \sim \text{Dir}(\beta_w)$
for each phrase $p$ do
    Choose aspect $a \sim \text{Mult}(\theta^A)$ and domain $d \sim \text{Mult}(\theta^D)$
    for each token $i = 1...|p_w|$ do
        draw $w_i \sim \text{Mult}(\phi_w^{d,a})$
    for each significant phrase $j = 1...|p_{sp}|$ do,
        draw $sp_j \sim \text{Mult}(\phi_{sp}^{d,a})$
    Draw venue $v \sim \phi_v^d$
    if $p_l$ exists then draw $p_l \sim \phi_l^a$
    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 $\mathcal{A}$ = [Technique(T), Application(A)]. We use the distributions ϕ_l and ϕ_r of the inferred aspects to set mapping $\mathcal{M}$ either to $M(T, A) = (1, 2)$ or $M(T, A) = (2, 1)$. 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 $\mathcal{RP}$, which are likely to appear as left relation phrases of Technique phrases, and right relation phrases of Application phrases, $\mathcal{M}$ is chosen to maximize the following objective:

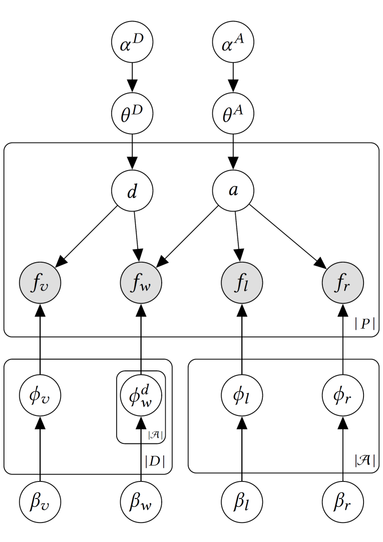


Figure 3: Graphical model for DomainPhraseType

$$\mathcal{M} = \arg\max \sum_{rp \in \mathcal{RP}} ([\phi_l(rp)]_{\mathcal{M}(\mathcal{T})} + [\phi_r(rp)]_{\mathcal{M}(\mathcal{A})})$$

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 ($\phi_{sp}^{d,a}$, $\phi_w^{d,a}$ and ϕ_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 ≥ 2:

- $(\phi_w^{d,a})_{t=T} \sim \omega\ (\phi_w^{d,a})_{t=T-1} + (1-\omega)\ \text{Dir}(\beta_w)$
- $(\phi_{sp}^{d,a})_{t=T} \sim \omega\ (\phi_{sp}^{d,a})_{t=T-1} + (1-\omega)\ \text{Dir}(\beta_w)$
- $(\phi_v^d)_{t=T} \sim \omega\ (\phi_v^d)_{t=T-1} + (1-\omega)\ \text{Dir}(\beta_v)$

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

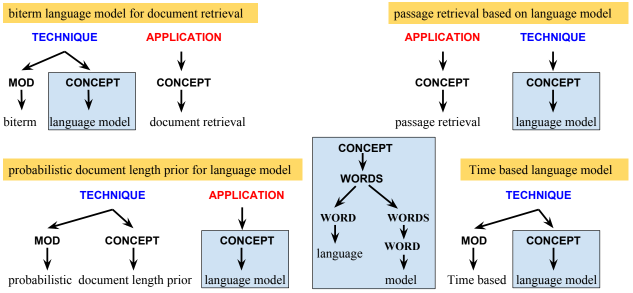


Figure 4: Illustration of adapted parse tree involving the adaptor CONCEPT to generate the phrase *language model*

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 probabilties 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 $\mathcal{G}$ is defined as a quintuple (N, W, R, S, θ). Given a finite set of terminals W, nonterminals N and start symbol S, $\mathcal{G}$ is given by a set of probabilistic grammar rules (R, θ) where R represents a set of grammar rules while θ 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, θ). 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 θ. 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 $\mathcal{PYG}$ is defined as,

- Finite set of terminals W, nonterminals N, rules R and start symbol S.
- Dirichlet prior α_A for the production probabilities θ_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^{th} 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^{th} customer chooses to sit at x_N with the following distribution (note that if he chooses an empty table, this is now the $K+1^{th}$ table),

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

where,

$$m_k = \#x_i, i \in \{1, ..., N-1\}, x_i = k$$

where δ_{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 $\mathbf{n} = \{n_1, ..., n_K\}$. This probability is

Phrase ⟶ Modifier ***Concept***
Phrase ⟶ ***Concept*** Modifier
Phrase ⟶ Modifier ***Concept*** Modifier
Phrase ⟶ Modifier

Concept ⟶ Words
Modifier ⟶ Words
Words ⟶ Word Words
Words ⟶ Word

Figure 5: Grammar rules to extract concepts and modifiers from typed phrases

given by,

$$P_{pyp}(\mathbf{n} \mid a, b) = \frac{\prod_{k=1}^{K}(b(k-1)+a)\prod_{j=1}^{m_k-1}(j-b)}{\prod_{i=0}^{n-1}(i+a)} \tag{1}$$

where K is the number of occupied tables and $(\sum_{i=1}^{K} n_i)$ is the total number of customers. In case of a $\mathcal{PYG}$, derivation trees are defined analogous to tables, and customers are instances of adapted non-terminals in the grammar. Thus when a new phrase is parsed, the most likely parse tree assigns the constituent non-terminals in the derivation to the popular tables, hence capturing significant concept mentions in our corpus.

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 $\mathbf{T}$ be the set of derivation trees used to derive P. The probability of $\mathbf{T}$ is then given by,

$$P(\mathbf{T} \mid \alpha, \mathbf{a}, \mathbf{b}) = \prod_{A \in N-C} p_{dir}(\mathbf{f_A}(\mathbf{T}) \mid \alpha_\mathbf{A}) \prod_{c \in C} p_{pyp}(\mathbf{n_c}(\mathbf{T}) \mid \mathbf{a_c}, \mathbf{b_c})$$

where $\mathbf{n_c}(\mathbf{T})$ represents the frequency vector of all adapted rules for adaptor c being observed in $\mathbf{T}$ and $\mathbf{f_A}(\mathbf{T})$ represents the frequency vector of all pcfg rules for nonterminal A being observed in $\mathbf{T}$. Here, $p_{pyp}(\mathbf{n} \mid a, b)$ is as given in Eqn. 1, while the dirichlet posterior probability $p_{dir}(\mathbf{f} \mid \alpha)$ for a given nonterminal is given by,

$$p_{dir}(\mathbf{f} \mid \alpha) = \frac{\Gamma(\sum\limits_{k=1}^{K} \alpha_k)}{\Gamma(\sum\limits_{k=1}^{K} f_k + \alpha_k)} \prod_{k=1}^{K} \frac{\Gamma(f_k + \alpha_k)}{\Gamma(\alpha_k)}$$

where $K = |R_A|$ is the number of PCFG rules associated with A, and variables $\mathbf{f}$ and α 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 $p(\mathbf{T} \mid \alpha, \mathbf{a}, \mathbf{b})$ 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

[†] DBLP dataset: https://datahub.io/dataset/dblp

Dataset	DBLP	ACL
Titles	188974	14840
Venues	79	50
Gold Standard titles	740	100
Gold Standard Technique	630	96
Gold Standard Application	783	108

Table 1: Dataset and Gold Standard statistics

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[1].

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 $(2000, 200, 2)$ as recommended in [23]. For PhraseType, we set parameters $\alpha = 50/|\mathcal{A}|$ and $\beta_w = \beta_l = \beta_r = 0.01$, while for DomainPhraseType, we set $\alpha_A = 50/|\mathcal{A}|$, $\alpha_D = 50/|D|$ and $\beta_w = \beta_l = \beta_r = \beta_v = 0.01$ and perform inferencing with collapsed gibbs sampling. Temporal parameter ω 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 α_A is set to 0.01.

Method \Dataset	DBLP			ACL		
	Prec	Rec	F1	Prec	Rec	F1
NP chunking	0.483	0.292	0.364	0.509	0.279	0.360
SegPhrase	0.652	0.376	0.477	0.784	0.451	0.573
PhraseType + Adaptor	**0.699**	**0.739**	**0.718**	**0.806**	**0.731**	**0.767**
DomainPhraseType + Adaptor:Mod	0.623	0.644	0.633	0.732	0.694	0.713
DomainPhraseType + Adaptor	0.698	0.736	0.716	0.757	0.709	0.732

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

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/

Method\Domain	**IR**			**ML**			**DM**		
	Prec	Rec	F1-Score	Prec	Rec	F1-Score	Prec	Rec	F1-Score
Bootstrapping + NP	0.437	0.325	0.373	0.4375	0.307	0.361	0.382	0.240	0.295
Bootstrapping + Segphrase	**0.717**	0.497	0.587	0.280	0.203	0.235	0.583	0.440	0.502
PhraseType + PCFG	0.444	0.487	0.465	0.374	0.390	0.382	0.364	0.434	0.396
PhraseType + Adaptor:Mod	0.599	0.669	0.632	0.513	0.522	0.517	0.537	0.657	0.591
PhraseType + Adaptor	0.712	**0.793**	**0.750**	**0.653**	**0.681**	**0.667**	**0.584**	**0.714**	**0.642**
	PL			**ALG**			**NW**		
Bootstrapping + NP	0.548	0.398	0.461	0.376	0.244	0.296	0.344	0.297	0.319
Bootstrapping + Segphrase	**0.617**	0.425	0.503	0.518	0.359	0.424	0.253	0.227	0.239
PhraseType + PCFG	0.478	0.478	0.478	0.378	0.436	0.405	0.145	0.158	0.151
PhraseType + Adaptor:Mod	0.576	0.569	0.572	0.506	0.583	0.542	0.402	0.445	0.422
PhraseType + Adaptor	0.604	**0.607**	**0.605**	**0.560**	**0.654**	**0.603**	**0.557**	**0.623**	**0.588**

Table 3: DBLP : Domain-specific results (Precision, Recall and F1 scores) - comparing PhraseType with baselines

Dataset	**Method**	**Application**			**Technique**			**Overall**		
		Prec	Rec	F1-Score	Prec	Rec	F1-Score	Prec	Rec	F1-Score
DBLP	Bootstrapping + NP	0.330	0.323	0.326	0.424	0.082	0.137	0.338	0.213	0.261
	Bootstrapping + Segphrase	0.418	0.432	0.425	0.431	0.053	0.094	0.419	0.253	0.316
	PhraseType + PCFG	0.369	0.381	0.375	0.370	0.425	0.396	0.370	0.402	0.385
	PhraseType + Adaptor	0.604	0.628	0.616	0.554	0.653	0.599	0.578	0.640	0.607
	DomainPhraseType + PCFG	0.412	0.430	0.421	0.397	0.456	0.424	0.405	0.443	0.423
	DomainPhraseType + Adaptor:Mod	0.603	0.618	0.610	0.523	0.598	0.558	0.563	0.609	0.585
	DomainPhraseType + Adaptor	**0.657**	**0.692**	**0.674**	**0.595**	**0.689**	**0.639**	**0.623**	**0.691**	**0.655**
ACL	Bootstrapping + NP	0.283	0.265	0.274	0.500	0.079	0.136	0.311	0.177	0.226
	Bootstrapping + Segphrase	0.655	0.582	0.616	0.625	0.170	0.267	0.648	0.387	0.485
	PhraseType + PCFG	0.326	0.316	0.321	0.341	0.341	0.341	0.333	0.328	0.330
	PhraseType + Adaptor	0.645	0.612	0.628	0.561	0.522	0.541	0.606	0.569	0.587
	DomainPhraseType + PCFG	0.412	0.408	0.410	0.413	0.375	0.393	0.412	0.392	0.402
	DomainPhraseType + Adaptor:Mod	0.680	0.673	0.676	0.616	**0.602**	**0.609**	0.650	0.639	0.645
	DomainPhraseType + Adaptor	**0.730**	**0.745**	**0.737**	**0.629**	0.579	0.603	**0.685**	**0.667**	**0.676**

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

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 E5345 (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)).

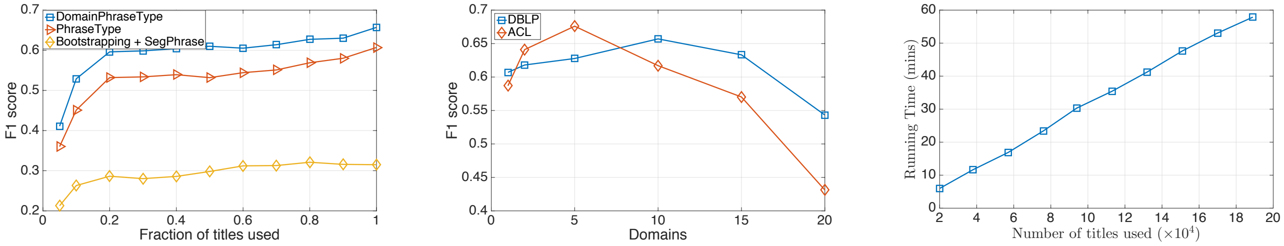
5.3 Case Studies

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.

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

Table 5: Modifiers for a few sample concepts

For a Technique concept such as *Neural Network*, modifiers such as *convolutional* and *recurrent* represent multiple variations of the technique proposed in different scenarios. The modifiers extracted for a concept provide a holistic perspective of the different variations in which the particular concept has been observed in research literature.



(a)

(b)

(c)

Figure 6: (a) Performance of *DomainPhraseType* on varying the corpus size in DBLP dataset (b) Performance of *DomainPhraseType* on varying the number of domains and (c) Runtime analysis for *DomainPhraseType* on the 2 corpora

Domains discovered in DBLP: In Table 6, we provide a visualization of the domains discovered by *DomainPhraseType* in the DBLP dataset. Table 6 shows the the most probable venues (ϕ_v) and a few popular concepts identified by *DomainPhraseType + Adaptor* for the articles 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 faceted 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 specfic concepts in a data-driven manner with adaptor grammars. While our

Domain #	1	2	3	4	5
Top venues ϕ_v	SIGIR, CIKM, IJCAI	ICALP, FOCS, STOC	OOPSLA, POPL, PLDI	CVPR, ICPR, NIPS	ACL, COLING, NAACL
Concepts	web search knowledge base search engine	complexity class cellular automaton model checking	flow analysis garbage collection program analysis	neural network face recognition image segmentation	machine translation natural language dependency parsing
Domain #	6	7	8	9	10
Top venues ϕ_v	ICDM, KDD, TKDE	ICC, INFOCOM, LCN	SIGMOD, ICDE, VLDB	ISAAC, COCOON, FOCS	WWW, ICIS, WSDM
Concepts	feature selection association rule time series	sensor network cellular network resource allocation	database system data stream query processing	planar graph efficient algorithm spanning tree	social network information system semantic web

Table 6: Domains discovered by *DomainPhraseType* in the DBLP dataset ($|D|$=10)

focus here has been to apply our algorithm on scientific titles to discover technique 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 grammatic 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 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative.

REFERENCES

[1] Steven P Abney. 1991. Parsing by chunks. In *Principle-based parsing*. Springer, 257–278.

[2] Eugene Charniak. 1997. Statistical Parsing with a Context-Free Grammar and Word Statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island.* 598–603.

[3] Ahmed El-Kishky, Yanglei Song, Chi Wang, Clare R Voss, and Jiawei Han. 2014. Scalable topical phrase mining from text corpora. *Proceedings of the VLDB Endowment* 8, 3 (2014), 305–316.

[4] Sonal Gupta and Christopher D. Manning. 2011. Analyzing the Dynamics of Research by Extracting Key Aspects of Scientific Papers. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011.* 1–9.

[5] Alon Halevy, Natalya Noy, Sunita Sarawagi, Steven Euijong Whang, and Xiao Yu. 2016. Discovering structure in the universe of attribute names. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 939–949.

[6] Hemant Ishwaran and Lancelot F James. 2003. Generalized weighted Chinese restaurant processes for species sampling mixture models. *Statistica Sinica* (2003), 1211–1235.

[7] Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2006. Adaptor Grammars: A Framework for Specifying Compositional Nonparametric Bayesian Models. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006.* 641–648.

[8] Mark Johnson, Thomas L Griffiths, and Sharon Goldwater. 2007. Bayesian Inference for PCFGs via Markov Chain Monte Carlo.. In *HLT-NAACL*. 139–146.

[9] Thomas Lin, Oren Etzioni, and others. 2012. No noun phrase left behind: detecting and typing unlinkable entities. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 893–903.

[10] Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. 2015. Mining quality phrases from massive text corpora. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 1729–1744.

[11] Christopher D. Manning and Hinrich Schütze. 2001. *Foundations of statistical natural language processing*. MIT Press.

[12] Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. 2013. Fine-grained Semantic Typing of Emerging Entities.. In *ACL (1)*. 1488–1497.

[13] Jim Pitman. 1995. Exchangeable and partially exchangeable random partitions. *Probability theory and related fields* 102, 2 (1995), 145–158.

[14] Xiaojun Quan, Chunyu Kit, Yong Ge, and Sinno Jialin Pan. 2015. Short and Sparse Text Topic Modeling via Self-Aggregation.. In *IJCAI*. 2270–2276.

[15] Dragomir Radev and Amjad Abu-Jbara. 2012. Rediscovering ACL discoveries through the lens of ACL anthology network citing sentences. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*. Association for Computational Linguistics, 1–12.

[16] Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2013. The ACL anthology network corpus. *Language Resources and Evaluation* 47, 4 (2013), 919–944. DOI: https://doi.org/10.1007/s10579-012-9211-2

[17] Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 147–155.

[18] Ellen Riloff and Jessica Shepherd. 1997. A corpus-based approach for building semantic lexicons. *arXiv preprint cmp-lg/9706013* (1997).

[19] Alan Ritter, Sam Clark, Oren Etzioni, and others. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1524–1534.

[20] Tarique Siddiqui, Xiang Ren, Aditya G. Parameswaran, and Jiawei Han. 2016. FacetGist: Collective Extraction of Document Facets in Large Technical Corpora. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016.* 871–880. DOI: https://doi.org/10.1145/2983323.2983828

[21] Yizhou Sun, Jie Tang, Jiawei Han, Manish Gupta, and Bo Zhao. 2010. Community evolution detection in dynamic heterogeneous information networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*. ACM, 137–146.

[22] Michael Thelen and Ellen Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 214–221.

[23] Chen-Tse Tsai, Gourab Kundu, and Dan Roth. 2013. Concept-based analysis of scientific literature. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013.* 1733–1738. DOI: https://doi.org/10.1145/2505515.2505613

[24] Roman Yangarber, Winston Lin, and Ralph Grishman. 2002. Unsupervised learning of generalized names. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1–7.

[25] Jianhua Yin and Jianyong Wang. 2014. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 233–242.

[26] Ke Zhai, Zornitsa Kozareva, Yuening Hu, Qi Li, and Weiwei Guo. 2016. Query to Knowledge: Unsupervised Entity Extraction from Shopping Queries using Adaptor Grammars. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 255–264.

[27] Chao Zhang, Guangyu Zhou, Quan Yuan, Honglei Zhuang, Yu Zheng, Lance Kaplan, Shaowen Wang, and Jiawei Han. 2016. Geoburst: Real-time local event detection in geo-tagged tweet streams. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 513–522.