Unsupervised concept based entity extraction from scientific titles

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

ABSTRACT
This paper studies the extraction and typing of entities from titles of academic literature, in order to gain a deeper understanding of their specific contributions and automate the construction of a problem-solution knowledgebase. To achieve this goal, we propose an unsupervised, domain independent, two phase algorithm to extract entity mentions and type them into appropriate concepts. In the first phase of our algorithm we propose a generative model which exploits textual and syntactic features to broadly segment titles and type them into concepts. In the second phase, we propose an unsupervised approach based on adaptor grammars to extract fine grained entities of interest without the need for any external resources or human effort, in a purely data driven manner. We analyze literature from diverse scientific domains and show significant gains over state-of-the-art concept extraction techniques. We also present an analysis and summarization of the knowledge base constructed by our algorithm.

CCS CONCEPTS
-Computer systems organization → Embedded systems; Redundancy; Robotics; Networks → Network reliability;

KEYWORDS
ACM proceedings, BRJX, text tagging

ACM Reference format:
DOI: 10.475/123_4

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 and topics in the literature, and understand their contributions via the key techniques and applications studied by them. The extracted information could be used to build a 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. Construction of a Problem-Solution knowledgebase can help answer questions such as - "what methods were developed to solve a particular problem?" and "how did these change over time?".

To achieve these objectives, it is necessary to extract the key entity mentions that are representative of a scientific article. We propose to use the titles of scientific articles to extract these mentions and type them into two primary concepts of interest, Techniques (Solutions) and Applications (Problems). We find that 95% of all paper titles that appeared in top IR/ML venues in the years 1970-2016, contain atleast 2 entity mentions. x% of these titles state both techniques and applications, whereas the remaining y% contain one of these concepts. Furthermore, titles are often structured to emphasize the most significant contributions of a scientific article.

Previous literature [13] has focused on the extraction of faceted entity mentions in the article text while 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. Titles provide a concise, yet accurate representation of the key concepts studied. Unlike the article text, titles often lack contextual information, and provide limited textual features which makes our problem challenging. Other techniques that focus on concept extraction [14] from the text typically first extract the candidate mentions by noun phrase chunking and then use bootstrapped features to categorize them into concepts. However, the use of general chunkers leads to inaccurate identification of entity mentions in specific domains such as scientific literature.

Our problem fundamentally differs from classic Named Entity Recognition techniques which focus on web resources via distant supervision [12] and natural language text [11]. Entity phrases corresponding to predefined categories such as person, organization, location etc are extracted using trigger words (pvt., corp., ltd., Mr./Mrs. etc.), grammar properties, syntactic structures such as dependency parsing, part of speech (POS) tagging and textual patterns. On the other hand, academic entities are not associated with any consistent trigger words and provide limited syntactic features. Titles lack context words and often follow syntax that is uncommon in general text, since they are required to be succinct. 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

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

∗Equal contribution
domain specific corpora to train supervised NER frameworks on academic text unlike general textual corpora such as news articles. This makes our problem fundamentally challenging and interesting to solve. The key requirements of our technique are as follows:

- It should not depend on specialist effort to annotate academic text or human curated external resources. The proposed technique must be unsupervised.

- Academic communities often evolve rapidly over time and several interdisciplinary publications cannot be concretely associated with a single domain. The technique must generalize to diverse domains by being data centric and learning from the data.

- It should not involve any significant parameter tuning. We do not require users to specify parameters such as length of entities, number of entities corresponding to each concept etc. apriori.

In this work, we propose a novel two step framework that satisfies the above requirements. Our first contribution is a concept-based generative model to explain the observation of different textual and syntactic features extracted from each title. This model is used to perform a coarse-grained segmentation of the title into multiple phrases and type these phrases into our two concept categories. Our second contribution is an unsupervised domain-independent algorithm to perform fine-grained typing of entities from the output of the previous step. Our model works based on adaptor grammars [6] - a non-parametric rule-based approach which is completely data-driven. We propose simple grammar rules that identify the key entities accurately and can provide additional information on different modifiers that occur in combination with the entities. To the best of our knowledge, ours is the first algorithm that can extract concept mentions from academic literature in an unsupervised setting. We conduct experiments on real-world datasets of scientific literature obtained from different domains. Our experimental results demonstrate significant performance improvements over existing concept extraction techniques. Finally, we also show the usefulness of the knowledgebase constructed by our framework by presenting a few case applications and applications.

2 RELATED WORK

The objective of this work is to automatically identify Technique and Application entities from academic paper titles in a completely unsupervised manner. This problem is quite different from the classic Named Entity Recognition (NER) problem and is significantly more challenging. NER techniques are based on noun phrase chunking which are usually trained on general-domain corpora like news articles and make use of various linguistic features, but do not work well on domain-specific corpora. Keyphrase extraction and phrase mining are aimed at extracting domain-specific keyphrases. However, they cannot type the extracted keyphrases into concepts and are hence not directly applicable in our problem.

This problem has been studied in earlier work in the weakly supervised setting while our method is completely unsupervised. Bootstrapping-based methods [2, 14] have been proposed for identifying important concepts in medical domain and scientific corpus which assume a seed list of high-quality entities for each concept. [2] use dependency parsing for each sentence to extract candidate mentions and apply a bootstrapping algorithm to extract three types of concepts - focus, technique, and application domain. [14] uses noun-phrase chunking to extract concept mentions and uses local textual features to annotate concept mentions iteratively. [14] is currently the state-of-the-art and is also the baseline for our work.

3 UNSUPERVISED PHRASE TYPING

Our approach consists of two primary phases. In the first phase, we segment paper titles into multiple phrases separated by relation phrases. Each segment is associated with a leading and following relation phrase, textual features including unigrams and frequent bigrams, and metadata associated with the paper title such as time of publication and conference or venue. These features aid us in the coarse typing of phrases into technique and application concepts. Topic models developed for modelling short text such as DMM [15] are insufficient in our problem owing to the nature of the text and the appearance of several common tokens in technique and application phrases. It is thus necessary to filter unigrams to retain only the most informative tokens.

3.1 PhraseType

Relation phrases often play consistent roles in paper titles. A relation phrase such as 'by applying' is likely to follow a problem phrase and lead a solution or technique phrase. However not all titles contain strongly indicative relation phrases. Furthermore, we find that 30% of all titles in our corpus contain no relation phrases and are composed of a single phrase. Thus it is necessary to build a combined model that can learn to appropriately type phrases. To this end, we propose a generic and flexible generative model (PhraseType) which models the generation of phrases jointly over all available evidence.

Phrases are assumed to be generated from a concept by joint draws of their respective textual features (filtered informative unigrams, significant phrases and higher n-grams) and relation phrases (left and right relation phrases, atleast one of which is available for each phrase). The corresponding graphical model is shown in Fig. 1.

\[
\begin{align*}
1) & \text{ Draw overall concept distribution in the corpus, } \theta \sim \text{Dir}(\alpha) \\
2) & \text{ For each concept } c, \\
& \text{ - Choose unigram distribution } \phi_w^c \sim \text{Dir}(\beta_w) \\
& \text{ - Choose significant phrase distribution } \phi_p^c \sim \text{Dir}(\beta_p) \\
& \text{ - Choose left relation phrase distribution } \phi_l^c \sim \text{Dir}(\beta_l) \\
& \text{ - Choose right relation phrase distribution } \phi_r^c \sim \text{Dir}(\beta_r) \\
3) & \text{ For each phrase } p \text{ present in the corpus, choose concept } c \sim \text{Mult}(\theta) \text{. Let } p_w, p_l, p_r \text{ represent the set of tokens in } p, p_l, p_r, \text{ the set of significant phrases in } p, \text{ and } p_l, p_r, \text{ the left and right relation phrases of } p \text{ respectively.} \\
& \text{ - For each token } i = 1 \ldots |p_w|, \text{ draw } w_i \sim \text{Mult}(\phi_w^c) \\
\end{align*}
\]
Unsupervised concept based entity extraction from scientific titles

**Figure 1: Pipeline of our concept extraction framework**

<table>
<thead>
<tr>
<th>PHRASE</th>
<th>UNIGRAM</th>
<th>SIGNIFICANT PHRASES</th>
<th>LEFT RP</th>
<th>RIGHT RP</th>
</tr>
</thead>
<tbody>
<tr>
<td>word embedding</td>
<td>word, embedding</td>
<td>word embedding</td>
<td>None</td>
<td>based</td>
</tr>
<tr>
<td>generalized</td>
<td>generalized</td>
<td>generalized</td>
<td>based</td>
<td>for</td>
</tr>
<tr>
<td>language</td>
<td>language, model</td>
<td>language, language model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>model</td>
<td>information</td>
<td>information retrieval</td>
<td>for</td>
<td>None</td>
</tr>
</tbody>
</table>

**Figure 2: Generative model for PhraseType**

- For each significant phrase \( j = 1 \ldots |p_{sp}| \), draw \( sp_j \sim \text{Mult}(\phi_{sp}) \)
- If a left relation phrase is present, draw \( p_l \sim \phi_l \)

**Figure 3: Generative model for DomainPhraseType**

- If a right relation phrase is present, draw \( p_r \sim \phi_r \)

We first generate the distributions over textual features and relation phrases for each concept based on the priors, and perform inference with MCMC collapsed gibbs sampling.
In addition to textual features and relation phrases, it is often necessary to study the influence of domain on techniques and applications. Modeling domain specific techniques and applications is likely to better disambiguate phrases into concepts. A simplification could be to use conferences as domains, however this would not capture interdisciplinary work well. Furthermore, most popular conferences often publish on several themes and diverse tracks. It would be beneficial to model domains as being composed of distributions over conferences. This enables our model to capture cross conference similarities and yet provide room to discover diverse papers within conferences and place them into an appropriate domain. We present our model DomainPhraseType to perform this task. We have not yet experimented with this model and plan to work on it in the near future.

3.2 DomainPhraseType

We study domains as being composed of distributions over concept specific textual features, which in our case corresponds to textual features of applications and techniques. Relation phrases are domain independent and play a consistent role with respect to concepts and are thus modeled independently. Additionally, venues could often encompass several themes and tracks, however they are fairly indicative of the broad domain of study. Thus, we model domains as simultaneous distributions over concept specific textual features, as well as venues. Every phrase is assumed to be jointly drawn from domains and concept via separate multinomial distributions. The textual features of the phrase are drawn from the corresponding concept distribution in the chosen domain and the relation phrases are drawn from the overall relation phrase distributions of the chosen concept. The generative process is as follows.

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

2) For each concept \( c \),
   - Choose left relation phrase distribution, \( \phi^L \sim \beta^L \).
   - Choose right relation phrase distribution, \( \phi^R \sim \beta^R \).

3) For each domain \( d \),
   i) For each concept \( c \),
      - Choose unigram distribution \( \phi^d_{w,c} \sim \text{Dir}(\beta^w) \).
      - Choose significant phrase distribution \( \phi^d_{sp,c} \sim \text{Dir}(\beta^p) \).
   ii) Draw a domain specific venue distribution \( \phi^d_{v,c} \sim \text{Dir}(\beta^v) \).

4) For each phrase \( p \) present in the corpus, choose concept \( c \sim \text{Mult}(\theta^C) \) and domain \( d \sim \text{Mult}(\theta^D) \). Let \( p_{w} \) represent 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.
   - For each token \( i = 1 \ldots |p_w| \), draw \( w_i \sim \text{Mult}(\phi^d_{w,c}) \)
   - For each significant phrase \( j = 1 \ldots |p_{sp}| \), draw \( sp_j \sim \text{Mult}(\phi^d_{sp,c}) \)
   - If a left relation phrase is present, draw \( p_L \sim \phi^L \) (Note that relation phrases are not domain specific)
   - If a right relation phrase is present, draw \( p_r \sim \phi^R \)

We first generate the distributions over textual features and syntactic features for each domain, concept pair, distribution over venues for each domain, and the left and right relation phrase distributions for each concept using the priors, and perform inference with MCMC collapsed gibbs sampling.

3.3 Temporal Dependencies

In addition to domains and concepts, it is necessary to model the temporal evolution of domains to better capture the variations that arise over time, with regard to the techniques and applications studied by articles published at various venues. It is beneficial to learn multiple models corresponding to varying time intervals, however there are commonalities between near time slices that can be explicitly incorporated into our model. Our primary objectives are two fold:

- The generative model must have sufficient flexibility to describe varying statistical information over different time periods.
- Evolution of the statistical features corresponding to a given domain and concept vary smoothly over time.

We therefore extend the above models in the time dimension. For the first timestamp (we discretize our dataset into timestamps of constant duration), the model follows the generative process described above on all phrases corresponding to paper titles belonging to the first timestamp. For subsequent timestamps the target phrases are modeled in a similar generative manner, however text and venue distributions \( \phi^d_{w,c}, \phi^d_{sp,c} \) and \( \phi^d_{v,c} \) are assumed to be described by a weighted mixture of the corresponding distribution learned in the previous timestamp, and the prior. Thus \( \forall T \geq 2 \):

\[
\begin{align*}
\phi^d_{w,c}(T) &= \omega \phi^d_{w,c}(T-1) + (1-\omega) \text{Dir}(\beta^w) \\
\phi^d_{sp,c}(T) &= \omega \phi^d_{sp,c}(T-1) + (1-\omega) \text{Dir}(\beta^p) \\
\phi^d_{v,c}(T) &= \omega \phi^d_{v,c}(T-1) + (1-\omega) \text{Dir}(\beta^v)
\end{align*}
\]

4 ADAPTOR GRAMMAR

Adaptor grammars are a class of probabilistic language models that generalize probabilistic context-free grammars (PCFGs) by augmenting the probabilistic rules of a PCFG to capture dependencies among successive uses. In addition, it has the advantage of being a non-parametric Bayesian model in contrast to PCFG that has parameters for each rule. It has been used widely in many linguistic tasks such as word segmentation [5] and morphological acquisition [4]. In this section, we will formally define PCFGs and Adaptor Grammars.

4.1 Probabilistic Context-free Grammars

Probabilistic Context-Free Grammars (PCFGs) are a probabilistic extension of Context Free Grammars [1] that define a probability

\[
\begin{align*}
\phi^d_{w,c}(T) &= \omega \phi^d_{w,c}(T-1) + (1-\omega) \text{Dir}(\beta^w) \\
\phi^d_{sp,c}(T) &= \omega \phi^d_{sp,c}(T-1) + (1-\omega) \text{Dir}(\beta^p) \\
\phi^d_{v,c}(T) &= \omega \phi^d_{v,c}(T-1) + (1-\omega) \text{Dir}(\beta^v)
\end{align*}
\]
Adaptor grammars jointly model the context in which rules are applied and the grammar in order to break this independence assumption. It is non-parametric model that specifies a set of adapted non-terminals called adaptors which allows it to dynamically learn more meaningful derivation trees and expand the rule set according to data. Adaptor Grammars are defined based on the the Pitman-Yor process [10] and are hence also known as Pitman-Yor Grammars (PYG). Formally, the Pitman-Yor Grammar (PYG) $\mathcal{A}$ 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 $A \in N$, $\theta_A \sim \text{Dir}(\alpha_A)$.
- Set of non-reursive adaptors $C \subseteq N$ with PYP parameters $a_A, b_A$ for each adaptor $A \in C$.

Next, we discuss how the Pitman Yor Grammar can be described using the popular Chinese Restaurant Process (CRP) [3].

### 4.3 Chinese Restaurant Process and PYG

The Chinese Restaurant Process (CRP) provides a commonly used view of the Pitman-Yor grammar (PYG), described by a scale parameter $\alpha$, discount factor $b$ and a base distribution $G_A$ for an adaptor $A \in C$.

The CRP assumes that dishes are served on a set of tables (the number of tables is not bounded), and each customer (adaptor) entering the restaurant chooses to sit at either one of the preoccupied tables or at a new table, dependent on the distribution of people on all the previous tables. The dishes served on the tables (which is equivalent to the different possible parse trees in our case) are drawn from the base distribution $G_A$.

The CRP follows a kind of rich get richer dynamics, i.e. new customers are more likely to sit at tables that have more people on them currently. This is equivalent to saying that we want the new derivation trees to be likely to use the same productions that have been used in previous trees in the corpus. The derivation trees are decided in a data driven manner.

Let us assume that when the $N^{th}$ customer enters the restaurant, the previous $N-1$ customers labeled $\{1, 2, \ldots, N-1\}$ have been seated on $K$ tables ($K \leq N - 1$). Let the $i^{th}$ customer be seated on table $x_i \in \{1, \ldots, K\}$. 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 $K+1^{st}$ table):

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

where,

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

Here, $\delta_{K+1}$ 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. This means that the probability distribution of any sequence of table assignments for customers depends only on the number of customers per table $n = \{n_1, \ldots, n_K\}$.
This probability is given by:

\[
P_{\text{pyp}}(n \mid a, b) = \frac{\prod_{j=1}^{K} (b(j-1) + a) \prod_{j=1}^{m_{j-1}} (j-b)}{\prod_{j=1}^{m_{j}} (t + a)}
\](1)

where \( K \) is the number of occupied tables and \( n \) is the total number of customers. Next, we discuss the inference of Adaptor Grammars using Markov Chain Monte Carlo (MCMC) techniques.

### 4.4 Inference

The objective of inference is to get a distribution over derivation trees given a collection of sentences as input. Let \( X \) be the collection of sentences and \( T \) be the set of all possible derivation trees. The joint distribution of \( T \) is given as:

\[
P(T \mid a, b) = \prod_{A \in N-C} p_{\text{dir}}(f_A(T) \mid \alpha_A) \prod_{A \in C} p_{\text{pyp}}(n_A(T) \mid a_A, b_A)
\]

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

\[
p_{\text{dir}}(n \mid \alpha) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\Gamma(\sum_{k=1}^{K} f_k + \alpha_k)} \prod_{k=1}^{K} \frac{\Gamma(f_k + \alpha_k)}{\Gamma(\alpha_k)}
\]

where \( K = |R(A)| \) is the number of PCFG rules associated with \( A \), and variables \( f \) and \( \alpha \) 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 \( p(T \mid a, b) \) over all derivation trees that yield \( x \).

Since this probability is intractable to compute, we use MCMC methods to sample \( t_i \sim p(t_i \mid x_i, T_{-i}) \). We view the PYG as a special kind of PCFG which adapts its production probabilities depending on its history, i.e. we construct a PCFG approximation \( G(N, W, R, S, \theta') \) which is static snapshot of the adaptor grammar given all derivation trees \( T_{-i} \). Given an adaptor grammar \( A = (N, W, R, S, \alpha, C, a, b) \), we define \( R' \) as:

\[
R' = R \cup \{A \rightarrow \text{yield}(x) : A \in C, x \in x_A\}
\]

where \( x_A \) is the set of all derivation trees observed in \( T_{-i} \) with root node labeled \( A \) and \( \text{yield}(x) \) is the terminal string or yield of the tree \( x \). These additional productions involving the adaptors rewrite directly to strings of terminal symbols, and their rule probability is given by:

\[
\theta'_{A \rightarrow \beta} = \left( \frac{m_{A} b_A + a_A}{n_A + a_A} \right) \left( \frac{f_{A \rightarrow \beta}(x_A) + \alpha_{A \rightarrow \beta}}{m_{A} + \sum_{A \rightarrow \beta \in R(A)} \alpha_{A \rightarrow \beta}} \right)
\]

\[
+ \sum_{x \in x_A, \text{yield}(x) = \beta} \left( \frac{n_x - b_A}{n_A + a_A} \right)
\]

where \( f_{A \rightarrow \beta}(x_A) \) is the frequency count of observing \( A \rightarrow \beta \) in set \( x_A \), \( n_x \) is the frequency count of observing a particular tree.

**Figure 5: Simple grammar used to extract entities and modifiers from typed phrases**

\( x \in x_A, m_A \) is the length of \( x_A \) which is the number of unique derivation trees in \( x_A \) and \( n_A \) is the frequency count of observing all trees rooted at \( A \). We can now use an efficient PCFG sampling procedure [7] on the \( G' \). A Metropolis-Hastings algorithm is used to generate draws from the conditional \( p(t_i \mid x_i, T_{-i}) \) by considering it as the proposal distribution. We accept the proposal \( t_i \) with probability:

\[
A(t_i, t'_i) = \min \left( 1, \frac{p(t_i \mid x_i, T_{-i}) \cdot p(t'_i \mid x_i, G')}{p(t'_i \mid x_i, T_{-i}) \cdot p(t_i \mid x_i, G')} \right)
\]

The inference procedure starts by randomly initializing derivation trees for all terminal string. At each iteration we pick a string \( x_i \) at random and the derivation tree \( t_i \). Construct the PCFG approximation \( G' \) from \( T_{-i} \). Sample a derivation tree from \( G' \), accept the proposal according to the above equation and adjust the associated counts with rules and grammars.

### 5 ENTITY EXTRACTION

The first step of our model returns typed phrases corresponding to our two different concepts. However, it may not be able to accurately identify the key entities present in the phrase. We expect key entities to have a high frequency in the corpus, but should occur in a particular pattern as part of each phrase. We use the adaptor grammar framework defined in the previous section to extract the entities and modifiers in each typed phrase. For instance, in a phrase like non-negative matrix factorization, matrix factorization is the entity and non-negative is the modifier.

We define a fairly simple set of rules for this grammar as shown in Figure 5.

### 6 EXPERIMENTS

We evaluate the effectiveness of our concept extraction framework in comparison to multiple strong baselines. We present several case studies that demonstrate the usefulness of our framework in analyzing the evolution scientific literature and communities. We briefly describe the dataset used and then discuss our experimental results.

#### 6.1 Dataset

We evaluate our approach on a set of paper titles from DBLP obtained from four areas of computer science - Information Retrieval (IR), Machine Learning (ML), Data Mining (DM) and Databases (DB). The corpus contains 46,315 titles published in related venues over the last 20 years. For evaluation, we use 400 gold truth titles that
Unsupervised concept based entity extraction from scientific titles

<table>
<thead>
<tr>
<th>Area/Domain</th>
<th>BM+ NP Chunking</th>
<th>BM+SegPhrase</th>
<th>PhraseType + PCFG</th>
<th>PhraseType + Adaptor</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR</td>
<td>0.333</td>
<td>0.5705</td>
<td>0.3268</td>
<td>0.7692</td>
</tr>
<tr>
<td>DM</td>
<td>0.2447</td>
<td>0.5593</td>
<td>0.3426</td>
<td>0.7342</td>
</tr>
<tr>
<td>ML</td>
<td>0.3859</td>
<td>0.3421</td>
<td>0.3157</td>
<td>0.7017</td>
</tr>
<tr>
<td>DB</td>
<td>0.3132</td>
<td>0.4096</td>
<td>0.3253</td>
<td>0.4698</td>
</tr>
</tbody>
</table>

Table 1: Overall Performance comparison on Entity recall on gold-truth data

<table>
<thead>
<tr>
<th>Area/Domain</th>
<th>BM+ NP Chunking</th>
<th>BM+SegPhrase</th>
<th>PhraseType + PCFG</th>
<th>PhraseType + Adaptor</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR</td>
<td>0.3214</td>
<td>0.4523</td>
<td>0.2738</td>
<td>0.7619</td>
</tr>
<tr>
<td>DM</td>
<td>0.2906</td>
<td>0.6976</td>
<td>0.4069</td>
<td>0.6744</td>
</tr>
<tr>
<td>ML</td>
<td>0.4561</td>
<td>0.122</td>
<td>0.3684</td>
<td>0.7543</td>
</tr>
<tr>
<td>DB</td>
<td>0.4</td>
<td>0.5181</td>
<td>0.2909</td>
<td>0.4698</td>
</tr>
</tbody>
</table>

Table 2: Performance comparison on Application Entity recall on gold-truth data

<table>
<thead>
<tr>
<th>Area/Domain</th>
<th>BM+ NP Chunking</th>
<th>BM+SegPhrase</th>
<th>PhraseType + PCFG</th>
<th>PhraseType + Adaptor</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR</td>
<td>0.3472</td>
<td>0.7083</td>
<td>0.375</td>
<td>0.8055</td>
</tr>
<tr>
<td>DM</td>
<td>0.1754</td>
<td>0.3508</td>
<td>0.2456</td>
<td>0.8245</td>
</tr>
<tr>
<td>ML</td>
<td>0.3157</td>
<td>0.5614</td>
<td>0.2631</td>
<td>0.6842</td>
</tr>
<tr>
<td>DB</td>
<td>0.1428</td>
<td>0.131</td>
<td>0.3982</td>
<td>0.4545</td>
</tr>
</tbody>
</table>

Table 3: Performance comparison on Technique Entity recall on gold-truth data

Figure 6: Variation in number of papers by year in our dataset of paper titles

Figure 7: Number of Application entities recognized as neural network and deep learning across different years

were manually annotated by experts. Figure 6 shows the number of papers by year in our dblp dataset.

6.2 Baselines

We compare our technique against multiple strong baselines.

- Bootstrapping + NP chunking [14]: This approach is bootstrapping based concept extraction approach and is currently the state-of-the-art technique for concept extraction in the scientific literature.
- Bootstrapping + Segphrase: We use Segphrase [8] which is a state-of-the-art phrase mining technique to extract candidate concept mentions and use the bootstrapping algorithm on the extracted phrases.

- PhraseType + PCFG: We use our phrase typing algorithm along with a PCFG grammar to extract typed entities.
- PhraseType + Adaptor Grammar: This is our final model that uses PhraseType and Adaptor Grammars to extract typed entities.

6.3 Evaluation

We compare the performance of our model with baselines on the labeled dataset of 400 titles. We use the evaluation metric as entity recall, i.e. the fraction of actual entities that are correctly identified by the model. The results obtained on Application and Technique Entities are shown in Table 2 and Table 3. The overall performance of these different models is shown in Table 1. We observe that
We perform a few case studies based on the entities extracted from XXX’17, XXX et al. In Figure 7, we observe that neural networks were popular a few years, and now we observe a resurgence again in recent times.

Figure 7: Number of Technique entities recognized as neural network and deep learning across different years

The results are shown in Figures 7 and 8. In Figure 7, we observe that neural networks were popular as an Application in early years, then their popularity decreased for a few years, and now we observe a resurgence again in recent times.

Figure 8: Number of Technique entities recognized as neural network and deep learning across different years

In the second case study in Figure 9, we compare the variation in popularity of two commonly used techniques in Machine Learning - Decision trees and Support vector machines. We clearly see the potential of our model to produce interesting analyses on the scientific literature.

Figure 9: Number of Technique entities recognized as decision tree and support vector across different years

An interesting observation can be seen from Figure 8 where the use of neural network as a Technique has surged in recent years, which is what we observe currently. While for deep learning, we can observe that it is being increasingly studied as an Application/Problem in recent times as it is becoming a more fundamental area of research.

7 CONCLUSIONS

REFERENCES


