Analyzing the Dynamics of Research by Extracting Key Aspects of Scientific Papers

Sonal Gupta
Department of Computer Science
Stanford University
sonal@cs.stanford.edu

Christopher D. Manning
Department of Computer Science
Stanford University
manning@cs.stanford.edu

Abstract

We present a method for characterizing a research work in terms of its focus, domain of application, and techniques used. We show how tracing these aspects over time provides a novel measure of the influence of research communities on each other. We extract these characteristics by matching semantic extraction patterns, learned using bootstrapping, to the dependency trees of sentences in an article's abstract. We combine this information with pre-calculated article-to-community assignments to study the influence of a community on others in terms of techniques borrowed and the 'maturing' of some communities to solve other problems. As a case study, we show how the computational linguistics community and its sub-fields have changed over the years with respect to their foci, methods used, and domain problems. For instance, we show that part-of-speech tagging and parsing have increasingly been adopted as tools for solving problems in other domains. We also observe that speech recognition and probability theory have had the most seminal influence.

1 Introduction

The evolution of ideas and the dynamics of a research community can be studied using the scientific articles published by the community. For instance, we may be interested in how methods spread from one community to another, or the evolution of a topic from a focus of research to a problem-solving tool. We might want to find the balance between technique-driven and domain-driven research within a field. Establishing such a rich insight of the development and progress of scientific research requires an understanding of more than just the "topics" of discussion or citation links between articles, which have been used in the previous work to study trend and impact of articles. As an example, to determine whether technique-driven researchers have greater or lesser impact, we need to be able to identify styles of work. To achieve this level of detail and to be able to connect together how methods and ideas are being pursued, it is essential to move beyond bag-of-words topical models. This requires an understanding of sentence and argument structure, and is therefore a form of information extraction.

To study the application *domains*, the *techniques* used to approach the domain problems, and the *focus* of scientific articles in a community, we propose to extract the following concepts from the articles

> **FOCUS**: an article's main contribution
> **TECHNIQUE**: a method or a tool used in an article, for example, expectation maximization and conditional random fields
> **DOMAIN**: an article's application domain, such as speech recognition and classification of documents.

For example, if an article concentrates on regularization in support vector machines and shows improvement in parsing accuracy, then its FOCUS and TECHNIQUE are regularization and support vector machines, and its DOMAIN is parsing. In contrast, an article that focuses on lexical features to improve parsing accuracy and uses support vector machines to train the model has FOCUS as lexical features and parsing, the TECHNIQUE being lexical features and support vector machines, and its DOMAIN still is parsing.[1] In this case, even though TECHNIQUEs and DOMAIN of both papers

[1] A community vs. a DOMAIN: a community can be as broad as computer science or statistics, whereas a DOMAIN is a specific application such as Chinese word segmentation.

are very similar, the FOCUS phrases distinguish them from each other. Note that a DOMAIN of one article can be a TECHNIQUE of another, and vice-versa. For example, an article that shows improvements in named entity recognition (NER) has DOMAIN as NER, however, an article that uses named entities as an intermediary tool to extract relations has NER as one of its TECHNIQUEs.

Our work uses information extraction patterns to extract the above three category phrases from articles. The phrases are extracted by matching semantic patterns in dependency trees of sentences. The input to the extraction system are some seed patterns (see Table 1 for examples) and it learns more patterns using a bootstrapping approach. Using a bag-of-words based approach, such as topic models, for this problem is not straightforward; true to their name, topic models generally only identify the topic or area of a paper (such as 'parsing' or 'speech recognition'), and neither provide nor label different cross-cutting aspects like techniques used or the application domain of the paper.

As a case study, we examine the articles published in the computational linguistics community. We study the influence of the community's sub-fields, such as parsing and machine translation, using the FOCUS, TECHNIQUE, and DOMAIN phrases extracted from the articles. We use the document collection from the ACL Anthology dataset[2] (Bird et al., 2008; Radev et al., 2009), since it has full text of papers available. To get the the sub-fields of the community, we use latent Dirichlet allocation (Blei et al., 2003) to find topics and label them by hand.[3] However, our general approach can be used to study any case of the influence of academic communities, including looking more broadly at the influence of statistics or economics across the social sciences.

We study how communities influence each other in terms of techniques that are reused, and show how some communities 'mature' so that the results they produce get adopted as tools for solving other problems. For example, the products of the part-of-speech tagging (POS) community have been adopted by many other communities that use POS tagging as an intermediary step, which is also confirmed in our results.

We also show the timeline of influence of communities. For example, our results show that formal computational semantics and unification-based grammars had a lot of influence in the late 1980s. The speech recognition and probability theory fields showed an upward trend of influence in the mid-1990s, and even though it has decreased in recent years, they still have a lot of influence on recent papers mainly due to techniques like expectation maximization and hidden Markov models. Therefore, our results show that overall they have been the most influential fields in the last two decades. Probability theory, unlike speech recognition, is traditionally not a separate sub-field of computational linguistics, but it is an important topic since many papers use and work on probabilistic approaches. We also show that the study of influence is different from studying popularity or hotness of communities, such as in (Griffiths and Steyvers, 2004; Hall et al., 2008), which is based on the expected number of papers published in the community in a given year.

Contributions We introduce a new categorization of key aspects of scientific articles, which is (1) FOCUS: main contribution, (2) TECHNIQUE: method or tool used, and (3) DOMAIN: application domain. We extract the aspects by matching semantic patterns to dependency trees and learn the patterns using bootstrapping. We propose a new definition of influence of a research community in terms of its key aspects adopted as techniques by the other communities. We present a case study on the computational linguistics community using the the three aspects extracted from its articles, both for verifying the results of our system, and for showing novel results for the dynamics and the overall influence of computational linguistics sub-fields. We introduce a dataset of abstracts labeled with the three categories.[4]

2 Related Work

While there is some connection to keyphrase selection in text summarization (Radev et al., 2002), extracting FOCUS, TECHNIQUE and DOMAIN phrases is fundamentally a form of information extraction, and there has been a wide variety of prior work in this area. Some work, including the seminal (Hearst, 1992), identified patterns (ISA relations) using hand-written rules, while other work has learned patterns over dependency graphs (Bunescu and Mooney, 2005). This work builds

[2] http://www.aclweb.org/anthology

[3] In this paper, we use the terms communities, sub-communities and sub-fields interchangeably.

[4] The dataset is available at http://cs.stanford.edu/people/sonal/fta for the research community.

on previous successful use of bootstrapping learning techniques in NLP (Yarowsky, 1995; Collins and Singer, 1999; Riloff and Jones, 1999); in its use of dependency patterns it is perhaps especially close to (Yangarber et al., 2000).

Topic models have been used to study popularity of communities (Griffiths and Steyvers, 2004), the history of ideas (Hall et al., 2008), and scholarly impact of papers (Gerrish and Blei, 2010). However, topic models do not extract detailed information from text as we do. Still, we use topic-to-word distributions from topic models as a way of describing sub-fields.

Demner-Fushman and Lin (2007) used hand written knowledge extractors to extract information, such as population and intervention, in their clinical question-answering system to improve ranking of relevant abstracts. Our categorization of key aspects is applicable for broader range of communities, and we learn the patterns by bootstrapping. Li et al. (2010) used semantic metadata to create a semantic digital library for chemistry and identified experimental paragraphs using keywords features. Xu et al. (2006) and Ruch et al. (2007) proposed systems, in clinical-trials and biomedical domain, respectively, to classify *sentences* of abstracts corresponding to categories such as introduction, purpose, method, results and conclusion to improve article retrieval by using either structured abstracts,[5] or hand-labeled sentences. Some summarization systems also use machine learning approaches to find 'key sentences'. The systems built in these papers are complimentary to ours since one can find relevant paragraphs or sentences and then extract the key aspects from them. Note that a sentence can have multiple phrases corresponding to our three categories, and thus classification of sentences will not be enough.

3 Approach

In this section, we explain how to extract phrases for each of the three categories (FOCUS, TECHNIQUE and DOMAIN) and how to compute the influence of communities.

3.1 Pattern Matching and Learning

From an article's abstract and title, we use the dependency trees of sentences and a set of semantic extraction patterns to extract phrases in each of

[5]Structure abstracts, which are used by some journals, have multiple sections such as PURPOSE and METHOD.

FOCUS	present → (direct object) work → (preposition_on) propose → (direct object)
TECHNIQUE	using → (direct object) apply → (direct object) extend → (direct object)
DOMAIN	system → (preposition_for) task → (preposition_of) framework → (preposition_for)

Table 1: Some examples of semantic extraction patterns that extract information from dependency trees of sentences. A pattern is of the form $T \rightarrow (d)$, where T is the trigger word and d is the dependency that the trigger word's node has with its successor.

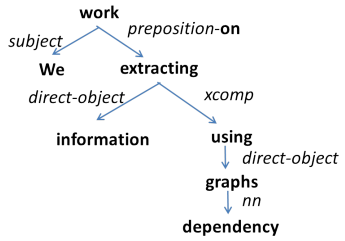


Figure 1: The dependency graph for '*We work on extracting information using dependency graphs*'. Our semantic patterns (shown in Table 1) will extract '*extracting information using dependency graphs*' as FOCUS, and '*dependency graphs*' as TECHNIQUE.

FOCUS, TECHNIQUE and DOMAIN categories. A dependency tree of a sentence is a parse tree that gives dependencies (such as direct-object, subject) between words in the sentence. Figure 1 shows the dependency graph for the sentence 'We work on extracting information using dependency graphs.' Each semantic pattern is of the form $T \rightarrow d$, where T is a trigger word (such as 'use', 'present') and d is a dependency (such as 'direct-object'). We start with a few handwritten patterns (some shown in Table 1) and learn more patterns automatically using a bootstrapping approach. We run an iterative algorithm that extracts phrases using semantic patterns and then learns new patterns from the extracted phrases. The details of each step are described below.

Extracting Phrases from Patterns A dependency tree matches a pattern $T \rightarrow (d)$, if (1) it contains T, and (2) the trigger word's node has a successor (dependent or granddependent upto 4 levels) whose dependency with its parent is d. In the rest of the paper, we call the subtree headed by the successor as the matched phrase-tree. We extract the phrase corresponding to the matched phrase-tree and label it with the pattern's category. For example, the dependency tree in Figure 1 matches the FOCUS pattern [*work → (preposition_on)*] and the TECHNIQUE pattern [*using → (direct-object)*]. Thus, the system labels the phrase corresponding to the phrase-tree headed

by 'extracting', which is 'extracting information using dependency graphs', with the category FOCUS, and similarly labels the phrase 'dependency graphs' as TECHNIQUE.

We have special rules for paper titles since authors usually include the main contribution of the paper in the title. We label the whole title as FOCUS if we are not able to extract a FOCUS phrase using the patterns. For titles from which we can extract a TECHNIQUE phrase, we label rest of the words (except for the trigger words) with DOMAIN. For example, for title 'Studying the history of ideas using topic models', our system extracts 'topic models' as TECHNIQUE using the pattern [*using* $\rightarrow$ *(direct-object)*], and then labels 'Studying the history of ideas' as DOMAIN.

Learning Patterns from Phrases After extracting phrases with patterns, we want to be able to construct and learn new patterns. For each sentence whose dependency tree has a subtree corresponding to one of the extracted phrases, we construct a pattern $T \rightarrow (d)$ by considering the ancestor (parent or grandparent) of the subtree as the trigger word T, and the dependency between the head of the subtree and its parent as the dependency d. The weighting of newly constructed patterns is done as follows. For a set of phrases (P) that extract a pattern (q), the weight of the pattern q for the category FOCUS is $\sum_{p \in P} \frac{1}{z_p} count(p \in \text{FOCUS})$, where z_p is the total frequency of the phrase p. Similarly, we get weights of the pattern for the other two categories. Note that we do not need smoothing since the phrase-category ratios are aggregated over all the phrases from which the pattern is constructed. After weighting all the patterns that have not been selected in the previous iterations, we select the top k patterns in each category (k=2 in our experiments). Table 3 shows some patterns learned through the iterative method.

3.2 Communities and their Influence

We define communities as fields or sub-fields that one wishes to study. To study communities using the articles published, we need to know which communities each article belongs to. The article-to-community assignment can be computed in several ways, such as by manual assignment, using metadata, or by text categorization of papers. In our case study, we use the topics formed by applying latent Dirichlet allocation (Blei et al., 2003) to the text of the papers by considering each topic as one community. In recent years, topic modeling has been widely used to get 'concepts' from text; it has the advantage of producing soft, probabilistic article-to-community assignment scores in an unsupervised manner. We combine these soft assignment scores with the phrases extracted in the previous section to score a phrase for each community and each category as follows. The score of a phrase p, which is extracted from an article a, for a community c and the category TECHNIQUE is calculated as

$$tScore(c, p, a) = \frac{1}{z_p} count(p \in \text{TECHNIQUE} \mid a) P(c \mid a, \theta) \quad (1)$$

where the function $P(c \mid a, \theta)$ gives the probability of a community (i.e., a topic) for the article a given the topic modeling parameters θ. The normalization constant for the phrase, z_p, is the frequency of the phrase in all the abstracts.

We define influence such that communities receive higher scores if they use techniques earlier than other communities do or produce tools that are used to solve other problems. For example, since *hidden Markov model* introduced by the speech recognition community and *part-of-speech tagging* tools built by the part-of-speech community have been widely used as techniques in other communities, these communities should receive higher scores than the nascent or not-so-widely-used ones. Thus, we define influence of a community based on the number of times its FOCUS, TECHNIQUE or DOMAIN phrases have been used as a TECHNIQUE in other communities. To calculate the overall influence of one community on another, we first need to calculate influence because of individual articles in the community, which is calculated as follows. The influence of community c_1 on another community c_2 because of a phrase p extracted from an article a_1 is

$$tInfl(c_1, c_2, p, a_1) = allScore(c_1, p, a_1) \sum_{\substack{a_2 \in D \\ y_{a_2} > y_{a_1}}} tScore(c_2, p, a_2) C(a_2, a_1) \quad (2)$$

where the function $allScore(c, p, a)$ is computed the same way as in Eq. 1, but by using $count(p \in \text{ALL} \mid a)$, where ALL means the union of phrases extracted in all three categories. The variable D is the set of all articles, and y_{a_2} means year of publication of the article a_2. The summation term computes the influence of the phrase p extracted

from the article a_1 on all the articles from the community c_2 published at a later date. The function $C(a_2, a_1)$ is a weighting function based on citations, whose value is 1 if a_2 cites a_1, and λ otherwise. If λ is 0, the system calculates influence based on just citations, which can be noisy and incomplete. In our experiments, we used λ as 0.5 since we want to study the influence even when an article does not explicitly cite another article. The technique-influence score of community c_1 on community c_2 in year y is computed by summing up the previous equation for all phrases (P) and for all articles in D. It is computed as

$$tInfl(c_1, c_2, y) = \sum_{p \in P} \sum_{\substack{a \in D \\ y_{a_1} = y}} tInfl(c_1, c_2, p, a) \quad (3)$$

Straightforwardly, the overall influence of community c_1 on the community c_2 and on all other communities is calculated as

$$tInfl(c_1, c_2) = \sum_y tInfl(c_1, c_2, y) \quad (4)$$

$$tInfl(c_1) = \sum_{c_2 \neq c_1} tInfl(c_1, c_2) \quad (5)$$

Next, we present a case study over the sub-fields of computational linguistics using the influence scores described above.

4 Experimental Setup

Dataset We studied the computational linguistics community from 1965 to 2009 using titles and abstracts of 15,016 articles from the ACL Anthology Network and the ACL Anthology Reference corpus (Bird et al., 2008; Radev et al., 2009). We found 52 pairs of abstracts that had more than 80% of words in common with each other, and thus while calculating the influence scores, we ignored the influence of earlier-published paper on the later-published paper in the pairs. We used the Stanford Parser (Marneffe et al., 2006) to generate dependency trees of sentences. For testing, we hand labeled 474 abstracts with the three categories to measure the precision and recall scores. For each abstract and each category, we compared the unique non-stop-words extracted from our algorithm to the hand labeled dataset. We calculated precision, recall measures for each abstract and averaged them to get the results for the dataset.

When extracting phrases from the matched phrase trees, we ignored tokens with part-of-speech tags as pronoun, number, determiner, punctuation or symbol, and removed all subtrees in the matched phrase trees that had either relative-clause-modifier or clausal-complement dependency with their parents since, even though we want full phrases, including these sub-trees introduced extraneous phrases and clauses. We also added phrases from the subtrees of the matched phrase trees to the set of extracted phrases.

We used 13 seed patterns for FOCUS, 7 for TECHNIQUE and 15 for DOMAIN. When constructing a new pattern, we ignored the ancestors that were not a noun or a verb since most trigger words are a noun or a verb (such as *use*, *constraints*). We also ignored conjunction, relative-clause-modifier, dependent (most generic dependency), quantifier-modifier, and abbreviation dependencies[6] since they either are too generic or introduced extraneous phrases and clauses.

Learning new patterns did not help in improving the FOCUS category phrases when tested over a hand labeled test set. It got relatively high scores when using just the seed patterns and the titles, and hence learning new patterns reduced the precision without any significant improvement in recall. Thus, we learned new patterns only for the TECHNIQUE and DOMAIN categories. We ran 50 iterations for both categories, which was chosen as a reasonable trade-off between pattern precision and recall based on some earlier pilot experiments. After extracting all the phrases, we removed common phrases that are frequently used in scientific articles, such as 'this technique' and 'the presence of', using a stop words list of 3,000 phrases. The list was created by taking the top most occurring 1 to 3 grams from 100,000 random articles with an abstract in the ISI web of knowledge database[7]. We ignored phrases that were either one character or more than 15 words long. In a step towards finding canonical names, we automatically detected abbreviations and their expanded forms from the full text of papers by searching for text between two parentheses, and considered the phrase before the parentheses as the expanded form (similar to (Schwartz and Hearst, 2003)). We got a high precision list by picking the top most occurring pairs of abbreviations and their expanded forms and created groups of phrases by merging all the phrases that use same abbreviation. We then changed all the phrases in the extracted phrases dataset to their canonical names.

[6] see (Marneffe et al., 2006) for details of dependencies

[7] www.isiknowledge.com

Paper Title	FOCUS	TECHNIQUE	DOMAIN
Studying the history of ideas using topic models	studying the history of ideas using topic	latent dirichlet allocation; topic; topic; unsupervised topic; historical trends; that all three conferences are converging in the topics	studying the history of ideas; topic; model of the diversity of ideas , topic entropy; probabilistic
A Bayesian Hybrid Method For Context-Sensitive Spelling Correction.	new hybrid method , based on bayesian classifiers; bayesian hybrid method for context-sensitive spelling correction	decision lists; bayesian; bayesian classifiers; ambiguous; part-of-speech tags; methods using decision lists; single strongest piece of evidence; spelling	context-sensitive spelling correction; for context-sensitive spelling correction; spelling

Table 2: Extracted phrases for some papers. The word 'model' is missing from the end of some phrases as it was removed during post-processing.

We also removed 'model', 'approach', 'method', 'algorithm', 'based', 'style' words and their variants when they occurred at the end of a phrase.

Baseline To compare against a non-information-extraction based baseline, we extracted all noun phrases, along with phrases from the sub-trees of the noun phrase trees, from the abstracts and labeled them with all the three categories. In addition, we labeled the titles (and their sub-trees) with the category FOCUS. We then scored the phrases with a tf-idf inspired measure, which was the ratio of the frequency of the phrase in the abstract and the sum of the total frequency of the individual words, and removed phrases that had the tf-idf measure less than 0.001 (best out of many experiments). We call this approach as 'Baseline tf-idf NPs'.[8]

To get communities in the computational linguistics literature, we considered the topics generated using the same ACL Anthology dataset by Bethard and Jurafsky (2010) as communities. They ran latent Dirichlet allocation on the full text of the papers to get 100 topics. We hand labeled the topics and used 72 of them in our study; the rest of them were about common words. When calculating the scores in Eq. 1, we considered the value of $P(c \mid a, \theta)$ to be 0 if it was less than 0.1.

5 Results and Discussion

Extraction

The total numbers of phrases extracted were 25,525 for FOCUS, 24,430 for TECHNIQUE, and 33,203 for DOMAIN. The total numbers of phrases after including the phrases extracted from subtrees of the matched phrase trees were 64,041, 38,220 and 46,771, respectively. Examples of phrases extracted from some papers are shown in Table 2.

[8]As discussed in Section 1, using an unsupervised or weakly-supervised bag-of-words based approach is not straightforward for identifying FOCUS, TECHNIQUE and DOMAIN of an article, and hence we do not compare against one.

TECHNIQUE	DOMAIN
model → (nn)	improve → (direct-object)
rules → (nn)	used → (preposition_for)
extracting → (direct-object)	evaluation → (nn)
identify → (direct-object)	parsing → (nn)
constraints → (amod)	domain → (nn)
based → (preposition_on)	applied → (preposition_to)

Table 3: Examples of patterns learned using the iterative extraction algorithm. The dependency 'nn' is the noun compound modifier dependency.

Approach	F_1	Precision	Recall
FOCUS			
Baseline tf-idf NPs	35.60	24.36	66.07
Seed Patterns	**55.29**	**44.67**	**72.54**
Inter-Annotator Agreement	53.33	50.80	56.14
TECHNIQUE			
Baseline tf-idf NPs	26.65	17.87	**52.41**
Seed Patterns	20.09	23.46	21.72
Iteration 50	**36.86**	**30.46**	46.68
Inter-Annotator Agreement	72.02	66.81	78.11
DOMAIN			
Baseline tf-idf NPs	30.13	19.90	**62.03**
Seed Patterns	25.27	**30.55**	26.29
Iteration 50	**37.29**	27.60	57.50
Inter-Annotator Agreement	72.31	75.58	69.32

Table 4: The precision, recall, and F_1 scores of each category for the different approaches. Note that the inter-annotator agreement is calculated on a smaller set.

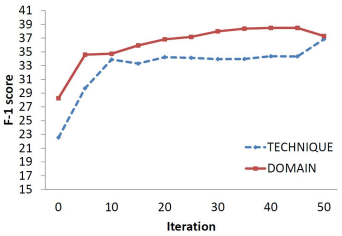
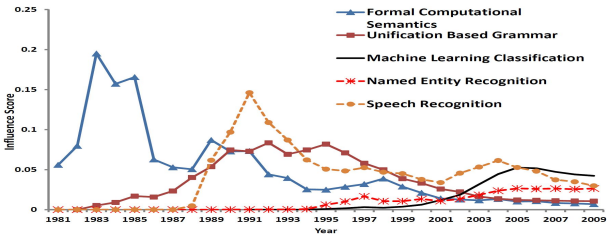


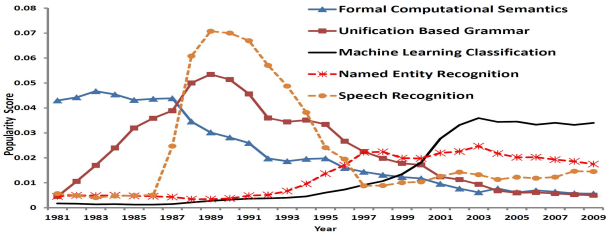
Figure 2: The F_1 scores for TECHNIQUE and DOMAIN categories after every five iterations. For reasons explained in the text, we do not learn new patterns for FOCUS.

Table 4 compares precision, recall, and micro-averaged F_1 scores for the three categories when we use: (1) only the seed patterns, (2) the combined set of learned and seed patterns, (3) the baseline, and (4) the inter-annotator agreement. We calculated inter-annotator agreement for 30 abstracts, where each abstract was labeled by 2 annotators,[9] and the precision-recall scores were calculated by randomly choosing one annotation as gold and another as predicted for each article. We

[9]The first author annotated 30 abstracts and two doctoral candidates in computational linguistics annotated 15 each.



(a) The influence of communities in each year.



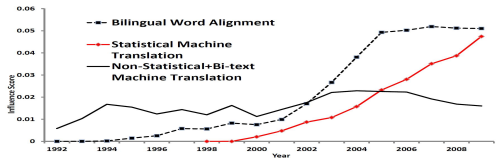
(b) Popularity of communities in each year.

Figure 3: The first figure shows influence scores of communities in each year. The second figure shows the popularity of each community in each year (see (Hall et al., 2008)), which is measured by summing up the article-to-topic scores for the articles published in that year. The scores are smoothed with weighted scores of 2 previous and 2 next years, and L1-normalized for each year. The scores are lower for all communities in late 2000s since the probability mass is more evenly distributed among many communities.

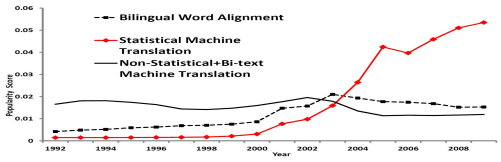
can see in the table that both precision and recall scores increase for TECHNIQUE because of the learned patterns, though for DOMAIN, precision decreases but recall increases. The recall scores for the baseline are higher as expected but the precision is very low. Three possible reasons explain the mistakes made by our system: (1) authors sometimes use generic phrases to describe their system, which were not annotated with any of the three categories in the test set but were extracted by the system (such as 'simple method', 'faster model', 'new approach'); (2) the dependency trees of some sentences were wrong; and (3) some of the patterns learned for TECHNIQUE and DOMAIN were low-precision but high-recall. Figure 2 shows the F_1 scores for TECHNIQUE and DOMAIN after every 5 iterations.

Influence

Table 5 shows the most influential communities overall (computed using Eq. 5) and their respective influential phrases that have been widely adopted as techniques by other communities. We can see that speech recognition is the most influential community because of the techniques like hidden Markov models and other stochastic methods it introduced in the computational linguistics literature, which shows that its long-term seeding influence is still present despite the limited recent



(a) The influence of communities in each year.



(b) Popularity of communities in a each year.

Figure 4: Comparing machine translation related communities in the same way as in Figure 3. The statistical machine translation community, which is a topic from the topic model, is more phrase-based.

popularity. Probability theory also gets a high score since many papers in the last decade have used stochastic methods. The communities part-of-speech tagging and parsing get high scores because they adopted some techniques that are used in other communities, and because other communities use part-of-speech tagging and parsing in the intermediary steps for solving other problems.

Figure 3(a) shows the change in a community's influence over time, and Figure 3(b) shows the change in its popularity. The popularity of a community is the sum of article-to-topic scores for the community topic and for all articles published in a given year.[10] The scores in both figures are normalized such that the total score for all communities in a year sum to one. Compare the relative scores of communities in Figure 3(a) with the relative scores in Figure 3(b). We can see influence of a community is different from the popularity of a community in a given year. As mentioned before, we observe that although influence score for speech recognition has declined in recent years, it still has a lot of influence, though the popularity of the community in recent years is very low. Machine learning classification has been both popular and influential in recent years. Named entity recognition's popularity has decreased since 2003, though its influence has either increased or remained same. Figure 4 compares the machine translation communities in the same way as we compare other communities in Figure 3. We can see that statistical machine translation (more phrase-based) community's popularity has steeply increased in the last 5 years, however, its influ-

[10] See (Hall et al., 2008) for more analysis. Note that this analysis uses just bag-of-words based topic models.

Community	Most Influential Phrases	Score
Speech Recognition (recognition, acoustic, error, speaker, rate, adaptation, recognizer, vocabulary, phone)	expectation maximization; hidden markov; language; contextually; segment; context independent phone; snn hidden markov; n gram back off language; multiple reference speakers; cepstral; phoneme; least squares; speech recognition; intra; hi gram; bu; word dependent; tree structured; statistical decision trees	1.35
Probability Theory (probability, probabilities, distribution, probabilistic, estimation, estimate, entropy, statistical, likelihood, parameters)	hidden markov; maximum entropy; language; expectation maximization; merging; expectation maximization hidden markov; natural language; variable memory markov; standard hidden markov; part of speech; inside outside; segmentation only; minimum description length principle; continuous density hidden markov; part of speech information; forward backward	1.31
Bilingual Word Alignment (alignment, alignments, aligned, pairs, align, pair, statistical, parallel, source, target, links, brown, ibm, null)	hidden markov; expectation maximization; maximum entropy; spectral clustering; statistical alignment; conditional random fields , a discriminative; statistical word alignment; string to tree; state of the art statistical machine translation system; single word; synchronous context free grammar; inversion transduction grammar; ensemble; novel reordering	1.2
POS Tagging (tag, tagging, pos, tags, tagger, part-of-speech, tagged, unknown, accuracy, part, taggers, brill, corpora, tagset)	maximum entropy; machine learning; expectation maximization hidden markov; part of speech information; decision tree; hidden markov; transformation based error driven learning; entropy; part of speech tagging; part of speech; variable memory markov; viterbi; second stage classifiers; document; wide coverage lexicon; using inductive logic programming	1.13
Machine Learning Classification (classification, classifier, examples, classifiers, kernel, class, svm, accuracy, decision, methods, labeled, vector, instances)	support vector machines; ensemble; machine learning; gaussian mixture; expectation maximization; flat; weak classifiers; statistical machine learning; lexicalized tree adjoining grammar based features; natural language processing; standard text categorization collection; pca; semisupervised learning; standard hidden markov; supervised learning	1.12
Statistical Parsing (parse, treebank, trees, parses, penn, collins, parsers, charniak, accuracy, wsj, head, statistical, constituent, constituents)	propbank; expectation maximization; supervised machine learning; maximumentropy classifier; ensemble; lexicalized tree adjoining grammar based features; neural network; generative probability; incomplete constituents; part of speech tagging; treebank; penn; 50 best parses; lexical functional grammar; maximum entropy; full comlex resource	0.92
Statistical Machine Translation (More-Phrase-Based) (bleu, statistical, source, target, phrases, smt, reordering, translations, phrase-based)	maximum entropy; hidden markov; expectation maximization; language; linguistically structured; ihmm; cross language information retrieval; ter; factored language; billion word; hierarchical phrases; string to tree; state of the art statistical machine translation system; statistical alignment; ist inversion transduction grammar; bleu as a metric; statistical machine translation	0.82

Table 5: The top most influential communities, along with the top most words that describe the communities obtained by the topic model, and the corresponding most influential phrases that have been widely used as techniques. The third column is the score of the community computed by Eq. 5.

Community	Communities that have influenced most (descending order)
Named Entity Recognition	Chunking/Memory Based Models; Discriminative Sequence Models; POS Tagging; Machine Learning Classification; Coherence Relations; Biomedical NER; Bilingual Word Alignment
Statistical Parsing	Probability Theory; POS Tagging; Discriminative Sequence Models; Speech Recognition; Parsing; Syntactic Theory; Clustering+DistributionalSimilarity; Chunking/Memory Based Models
Word Sense Disambiguation	Clustering + DistributionalSimilarity; Machine Learning Classification; Dictionary Lexicons; Collocations/Compounds; Syntax; Speech Recognition; Probability Theory

Table 6: The community in the first column has been influenced the most by the communities in the second column. The scores are calculated using Eq. 4

ence has increased at a slower rate. On the other hand, the influence of bilingual word alignment (the most influential community in 2009) has increased during the same period, mainly because of its influence on statistical machine translation. The influence of non-statistical machine translation has been decreasing recently, though slower than its popularity. Table 6 shows the communities that have the most influence on a given community (the list is in descending order of scores by Eq. 4).

6 Future Directions

We are working towards incorporating the date of publication of the articles to learn better patterns to increase precision and recall of the system. We are also exploring ways to use our system for studying citation and co-authorship networks. We plan to study the dynamics and impact of broader communities like biology, statistics and the social sciences. The approach can also be used to study innovation in interdisciplinary research, since we can track if interdisciplinary research results in applying old techniques from one community to solve problems in other community, or if it results in the evolution of better suited techniques.

7 Conclusions

This paper presents a framework for extracting detailed information from scientific articles, such as main contributions, tools and techniques used, and domain problems addressed, by matching semantic extraction patterns in dependency trees. We start with a few hand written seed patterns and learn new patterns using a bootstrapping approach. We use this rich information extracted from articles to study the dynamics of research communities and to define a new way of measuring influence of one research community on another. We present a case study on the computational linguistics community, where we find the *influence* of its sub-fields and observed that speech recognition and probability theory have had the most seminal influence.

References

Steven Bethard and Dan Jurafsky. 2010. Who should I cite: learning literature search models from citation behavior. In *Proceedings of the Conference on Information and Knowledge Management*.

Steven Bird, Robert Dale, Bonnie J. Dorr, Bryan Gibson, Mark T. Joseph, Min yen Kan, Dongwon Lee, Brett Powley, Dragomir R. Radev, and Yee Fan Tan. 2008. The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Conference on Language Resources and Evaluation*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Dina Demner-Fushman and Jimmy Lin. 2007. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*.

Sean M. Gerrish and David M. Blei. 2010. A language-based approach to measuring scholarly impact. In *Proceedings of the International Conference on Machine Learning*.

T. L. Griffiths and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*.

David Hall, Daniel Jurafsky, and Christopher D. Manning. 2008. Studying the history of ideas using topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Conference on Computational linguistics*.

Na Li, Leilei Zhu, Prasenjit Mitra, Karl Mueller, Eric Poweleit, and C. Lee Giles. 2010. OreChem ChemXSeer: a semantic digital library for chemistry. In *Proceedings of the Joint Conference on Digital libraries*.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Conference on Language Resources and Evaluation*.

Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the special issue on summarization. *Computational Linguistics*.

Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. 2009. The acl anthology network corpus. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Patrick Ruch, Clia Boyer, Christine Chichester, Imad Tbahriti, Antoine Geissbhler, Paul Fabry, Julien Gobeill, Violaine Pillet, Dietrich Rebholz-Schuhmann, Christian Lovis, and Anne-Lise Veuthey. 2007. Using argumentation to extract key sentences from biomedical abstracts. *International Journal of Medical Informatics*.

Ariel S. Schwartz and Marti A. Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text.

Rong Xu, Kaustubh Supekar, Yang Huang, Amar Das, and Alan Garber. 2006. Combining text classification and hidden markov modeling techniques for categorizing sentences in randomized clinical trial abstracts. *Proceedings of the Association of Moving Image Archivists Annual Symposium*.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Unsupervised discovery of scenario-level patterns for information extraction. In *Proceedings of the Conference on Applied Natural Language Processing*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the Association for Computational Linguistics*.