AKMiner: Domain-Specific Knowledge Graph Mining from Academic Literatures

Shanshan Huang and Xiaojun Wan*

Institute of Computer Science and Technology,
The MOE Key Laboratory of Computational Linguistics, Peking University,
Beijing 100871, China
{huangshanshan2010,wanxiaojun}@pku.edu.cn

Abstract. Existing academic search systems like Google Scholar usually return a long list of scientific articles for a given research domain or topic (e.g. "document summarization", "information extraction"), and users need to read volumes of articles to get some ideas of the research progress for a domain, which is very tedious and time-consuming. In this paper, we propose a novel system called AKMiner (Academic Knowledge Miner) to automatically mine useful knowledge from the articles in a specific domain, and then visually present the knowledge graph to users. Our system consists of two major components: a) the extraction module which extracts academic concepts and relations jointly based on Markov Logic Network, and b) the visualization module which generates knowledge graphs, including concept-cloud graphs and concept relation graphs. Experimental results demonstrate the effectiveness of each component of our proposed system.

Keywords: Knowledge graph generation, academic knowledge extraction, academic literature mining, Markov logic, AKMiner.

1 Introduction

Academic literatures offer scientific researchers knowledge about current academic progress as well as history in a specific research domain (e.g. "document summarization", "information extraction"). By reading scientific literatures, the beginners grasp basic knowledge of a research domain before in-depth study, and experienced researchers conveniently obtain the information of recent significant progress. However, relevant academic information is usually overloaded, and researchers often find an overwhelming number of publications of interests. Digital libraries offer various database querying tools, and Internet search companies have developed academic search engines. Typical academic search engines like *Google Scholar*, *Microsoft Academic Search* and *CiteSeer*, all achieve good retrieval performance.

However, most of the academic search systems simply return a list of relevant articles by matching keywords or author's information. Usually, there are quite a lot of articles for a specific topic, and it is hard for researchers to have a quick glimpse on the whole structure of knowledge, especially for the beginners.

* Corresponding author.

X. Lin et al. (Eds.): WISE 2013, Part II, LNCS 8181, pp. 241–255, 2013.

Another way to catch the development of a research domain or topic is to read review papers, such as survey papers published in ACM Computing Surveys every year. However, there are usually only a few high-quality review papers available for most research domains. Besides, the publishing cycle of review papers is relatively long, compared to fast updating of research achievements. Moreover, review papers are usually very long, and the knowledge is embedded in texts, which often fails to show the knowledge structure visually and vividly.

To help researchers relieve the burden of tedious paper reading, and acquire information about the recent achievements and developments quickly, we propose a novel system called AKMiner (Academic Knowledge Miner) to extract academic concepts and relations from academic literatures and generate knowledge graphs for a given research domain or topic. Hence, researchers can quickly get a basic vision of a research domain and learn the latest achievements and developments directly.

Our AKMiner system consists of two phases: a) extraction of academic concepts and relations, and b) academic knowledge graph generation. For the first step, Markov Logic Network (MLN) is applied to build a joint model for extracting academic concepts and their relations from literatures simultaneously. A concept cloud graph and a concept relation graph are then generated to visually present domain-specific concepts and relations, respectively. For concept cloud graph generation, the PageRank algorithm [19] is applied to calculate the importance scores of different concepts in a domain. The more important a concept is, the bigger it is displayed in the graphs. Experiments are conducted on datasets in four domains, and the training data and test data are from different domains. Evaluation results show that our proposed approach is more effective than several baseline methods (e.g. support vector machine (SVM), conditional random fields (CRF), C-Value) for knowledge extraction. Case studies show several good characteristics of the generated knowledge graphs.

The contributions of our study are summarized as follows:

- We propose a novel AKMiner system to mine knowledge graphs for a research domain, which can be used for enhancing existing academic search systems.
- We propose a joint model based on Markov Logic Network to extract academic concepts and their relations.

Experimental results on four datasets and cases of knowledge graphs show the effectiveness of our proposed system.

2 Related Works

2.1 Information Extraction

Information extraction (IE) techniques have been widely investigated for various purposes in the text mining and natural language processing fields. Typical IE tasks include named entity recognition (NER), relation detection and classification, and event detection and classification. State-of-the-art NER systems usually formulate NER as a sequence labeling problem, and employ various discriminative structured prediction models (e.g. hidden Markov model, maximum-entropy Markov model, CRF) to resolve it. Relation detection and classification aims to extract relations among entities and classify them into different categories. Many statistical techniques have been investigated to predict entity relations such as [32]. Recently, joint models

have been proposed to extract entities and relations simultaneously [20], [21], which achieve superior performance to pipeline models. Event detection and classification aims to mine significant events and group them into relevant topics, benefiting more discussions and comparisons within and crossing topics.

Term extraction or terminology extraction is a subtask of information extraction, which aims to extract multi-word expressions from a large corpus. Different methodologies for automatic term extraction have been investigated, including linguistic, statistical and hybrid approaches [5]. Linguistic approaches basically identify terms by using some heuristic rules or patterns [8]. Statistical approaches usually rank candidate terms according to a criterion, which is able to distinguish among true and false terms and give higher confidence to the better terms [3], [7], [10]. Hybrid approaches usually combine linguistic and statistical approaches into a two-stage framework [7], [15]. Keyphrase extraction is a very similar task with term extraction. Most keyphrase extraction methods first extract candidate phrases with natural language processing techniques, and then rank the candidate phrases and select the final keyphrases with supervised or unsupervised algorithms [14], [18], [30].

2.2 Academic Literature Mining

In recent years, various text mining techniques have been investigated in the academic domain. Such techniques include metadata extraction [11], [13], paper summarization [1], [23], survey generation [2], [23], [31], literature search [9], [17], paper recommendation [28], and trend visualization [6], [25], [29]. In particular, [29] extracts elemental technologies through the structure of research paper's title and analyzes technical trend in any research fields. [6] reports an effort to integrate statistics, text analytics, and visualization in a multiple coordinated window environment to support rapid understanding of scientific paper collections. [25] creates metro maps using metrics of influence, coverage, and connectivity for scientific literature, and shows the relations between papers. However, they all do not present fine-grained knowledge structure from paper content directly.

2.3 Markov Logic Networks

MLN is a powerful representation for statistical relational learning, and it has been applied in a few tasks in the area of information extraction. For example, [2] uses Markov Logic Network to extract database records from text or semi-structured sources. [26] proposes a system utilizing MLN for entity resolution, and [22] implements it into joint unsupervised coreference resolution. Besides, [27] uses MLN to discover social relationships in consumer photo collections.

3 Overview

3.1 System Overview

The purpose of our AKMiner system is to generate knowledge graphs for academic domains, which is useful for researchers to get useful information quickly and

visually. Given a set of literatures in a specific domain, academic knowledge graphs can be built through the flow in Figure 1.

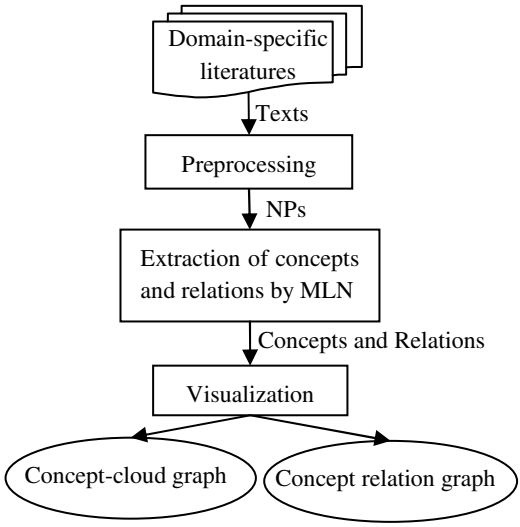

Fig. 1. Framework of AKMiner system

The framework of our proposed system consists of two main procedures: a) the extraction module which extracts academic concepts and relations, and b) the visualization module which visualizes knowledge graphs. Prior to the two main procedures, a preprocessing step is taken, as described in Section 6. To make inputs for MLN, we use a chunking tool to get noun phrases (NPs) from academic literatures and build the candidate set by some preprocessing. Then, we extract useful academic concepts and relations among them with our proposed joint model. Finally, knowledge graphs, including concept-cloud graph and concept relation graph, are generated based on the extracted concepts and relations.

3.2 Definition of Concepts and Relations

In this section, we define academic concepts and relations used in our system. To describe an academic domain, we often present the main tasks in the domain, and summarize the main methods applied to solve the tasks. Hence, in this study, we focus on two kinds of academic concepts: *Task* and *Method*.

The *Task* concepts are specific problems to be solved in academic literatures, including all concepts related to tasks, subtasks, problems and projects, like "machine translation", "document summarization", "query-focused summarization", etc.

The *Method* concepts are defined as ways to solve specific *Tasks*, including all concepts describing algorithms, techniques, models, tools and so on, like "Markov logic", "CRFs", "heuristic-based algorithm", and "XML-based tool".

The relations extracted by our system include two kinds of relations: the relations between the two kinds of academic concepts (i.e. *Method-Task* relations), and the relations within the same kind of academic concepts (i.e. *Method-Method* relations and *Task-Task* relations). The first kind of relations are formed when a *Method* is applied to a referred *Task* (e.g., "extractive method" for "document summarization"). The latter kind of relations between *Methods* or between *Tasks* are formed by dependency, evolution and enhancements (e.g., "Markov model" and "hidden Markov model", "single document summarization" and "multi-document summarization").

4 Markov Logic Networks

Markov Logic Network (MLN) [24] is a probabilistic logic model which combines the idea of Markov network with first-order logic. A first-order knowledge base (KB) is a set of formulas in first-order logic, in which the predicates and functions are used to describe properties and relations among the objects. Particularly, a function (e.g., Anna = MotherOf(Bob)) represents a mapping from objects to objects, while a predicate represents relations among objects or some attributes (e.g., Friends(Jim, Bob)). In a first-order KB, if a case violates even one formula, it has zero probability.

The basic idea of Markov logic is to soften these hard constraints with associated weights, so that they can be violated with some penalty.

Formally, a Markov Logic Network L is a set of pairs (F_i, w_i), where F_i, is a formula in first-order logic, and w_i is the weight of formula F_i. A Markov Logic Network specifies a probability distribution over the set of possible worlds χ as follows,

$$P(X = x) = \frac{1}{Z} \cdot exp\left(\sum_i w_i \cdot n_i(x)\right)$$

where $n_i(x)$ is the number of true groundings of F_i in possible worlds χ, and Z is the normalization constant, given by $Z = \sum_{x \epsilon \chi} exp(\sum_i w_i \cdot n_i(x))$.

5 MLN for Extraction of Concepts and Relations

In this section, we describe the construction of Markov Logic Network for concepts and relations extraction from academic literatures. Particularly, the attributes and relations of concepts are represented by predicates, and rules impose certain constraints over those predicates. We now describe our proposed approach in detail.

5.1 Concepts and Predicates

Our goal is to extract *Method* and *Task* concepts. For sake of predicates formulation, we add another kind of concepts, *Other* concept, not belonging to *Method* nor *Task*.

We define two kinds of predicates: query predicates and evidence predicates. The values of query predicates are unknown and need to be inferred. Here, the query predicates are the categories of concepts and the relations between them, including:

- **Method(concept)**: It indicates that the concept is a *Method*.
- **Task(concept)**: It indicates that the concept describes a *Task*.
- **Relation(concept$_1$,concept$_2$)**: It indicates that there is a relation between a *Method* and a *Task*. The first concept is a *Method* and the second one is a *Task*.
- **Relation_m(concept$_1$,concept$_2$)**: It indicates a relation between two *Methods*.
- **Relation_t(concept$_1$,concept$_2$)**: It indicates a relation between two *Tasks.*

The evidence predicates can be any features extracted from the inputs, and the values of them are already known. We represent the evidence predicates as follows:

- **Key_m(concept)**: A concept contains a keyword related to *Method* concept. For example, “graph-based method” contains *Method* related keyword “method”.
- **Key_t(concept)**: A concept contains a keyword related to *Task* concept.

- **Key_m_outside(concept)**: A keyword related to *Method* is detected around the concept in a text. For example, in the text "… we apply CRF in sequence annotation …", the keyword "apply" appears before concept "CRF".
- **Key_t_outside(concept)**: A keyword related to *Task* is detected around the concept in a text. For example, the keywords "task of" occurs before *Task* concept "paper summarization" in the text "… task of paper summarization …".
- **Neighbor(concept$_1$,concept$_2$)**: Two concepts appearing closely are neighbors. We define "closely" within three sentences in our experiments.
- **Contain(concept$_1$,concept$_2$)**: This predicate is true only in two cases. One is that concept c_1 only contains one more suffix than concept c_2, such as "CRF model" and "CRF". Another case is that the last words of two concepts are the same, such as concept "single document summarization" and "document summarization".
- **Apposition(concept$_1$,concept$_2$)**: It indicates that the concepts are appositional in a text. We define the appositional format as "concept$_1$ and (or) concept$_2$".

Note that in our experiments, all the keywords are collected and summarized manually. We investigate through reading numerous articles and collect four lists of keywords for Key_m, Key_t, Key_m_outside and Key_t_outside, respectively. The keywords for Key_m/t only contain nouns (17 words for *Method*, and 10 for *Task*), such as "algorithm", "method", "model" for *Method*, and "project", "problem" for *Task*. The keywords for Key_m/t_outside contain nouns and verbs (60 for *Method* and 31 for *Task*), such as "propose", "present", "describe", etc. These keywords are independent of domains, and can be utilized in any domains.

5.2 Rules in MLN

Hard Rules

Hard rules describe the hard constraints that should always hold true. These rules are given a prior weight larger than other rules.

Non-overlapping Rules

We make the rule that three categories of concepts (*Task*, *Method* and *Other*) do not overlap. That is to say, a concept only belongs to one of the categories.

$$Task(c) \Rightarrow !Method(c) \wedge !Other(c) \quad (1)$$

$$Method(c) \Rightarrow !Task(c) \wedge !Other(c) \quad (2)$$

$$Other(c) \Rightarrow !Task(c) \wedge !Method(c) \quad (3)$$

Rules from Definition.

These rules are from the definition of three query predicates indicating relations.

$$Relation(c_1, c_2) \Rightarrow Method(c_1) \wedge Task(c_2) \quad (4)$$

$$Relation\_m(c_1, c_2) \Rightarrow Method(c_1) \wedge Method(c_2) \quad (5)$$

$$Relation\_t(c_1, c_2) \Rightarrow Task(c_1) \wedge Task(c_2) \quad (6)$$

Soft Rules

These rules describe constraints that we expect to be usually true, but not all the time.

Neighbor-based Rules

We assume that two neighbor concepts probably have a relation.

$$Neighbor(c_1, c_2) \wedge Method(c_1) \wedge Task(c_2) \Rightarrow Relation(c_1, c_2) \quad (7)$$

$$Neighbor(c_1, c_2) \wedge Method(c_1) \wedge Method(c_2) \Rightarrow Relation\_m(c_1, c_2) \quad (8)$$

$$Neighbor(c_1, c_2) \wedge Task(c_1) \wedge Task(c_2) \Rightarrow Relation\_t(c_1, c_2) \quad (9)$$

Keyword-based Rules

We consider keywords as important clues to extract academic concepts. For example, a concept with the word "algorithm" as suffix is probably a *Method*, and a concept with "problem" as suffix is likely to be a *Task*. But sometimes this rule is not true. For instance, "efficient method" is not a useful concept. So keyword-based rules are soft.

$$Key\_m(c) \Rightarrow Method(c) \quad (10)$$

$$Key\_t(c) \Rightarrow Task(c) \quad (11)$$

In addition, the words around concept phrases also offer much useful information, such as the words "propose", "demonstrate", and "present". A concept with this kind of keywords appearing around probably belongs to the related concept category. The rules are as follows.

$$Key\_m\_outside(c) \Rightarrow Method(c) \quad (12)$$

$$Key\_t\_outside(c) \Rightarrow Task(c) \quad (13)$$

Containing-based Rules

From texts, we find that if the predicate *Contain* (c_1, c_2) is true, c_1 likely contains more modifiers than c_2. If c_1 is a *Task*, c_2 tends to be a *Task*, too. For example, *Contain("Spanisah to English MT", "MT") ^ Task("Chinese to English MT") ⇒ Task("MT")*. In this case, c_1 helps to determine the category of c_2. But if c_1 is a *Method*, we cannot perform the inference. For example: *Contain("phrase-based MT", "MT") ^ Method("phrase-based MT") ⇏ Method("MT")*.

However, if c_2 is a *Method* concept, c_1 is probably a *Method* concept. For instance, *Contain ("Hidden Markov model", "Markov model") ^ Method ("Markov model") => Method ("Hidden Markov model")*. On the other hand, if c_2 is a *Task* concept, the inference is unreasonable. For example, *Contain ("phrase-based statistical machine translation", "machine translation") ^ Task ("machine translation") ⇏ Task ("phrase-based statistical machine translation")*. In all, we have the formulas below.

$$Contain(c_1, c_2) \wedge Task(c_1) \Rightarrow Task(c_2) \quad (14)$$

$$Contain(c_1, c_2) \wedge Method(c_2) \Rightarrow Method(c_1) \quad (15)$$

Apposition Rules

Apposition information can also be used for extraction of concepts. Given the predicate *Apposition* (c_1, c_2), we add a rule that if two concepts are appositive, they likely belong to the same category. The rule is built below:

$$Apposition(c_1, c_2) \Rightarrow \big(Method(c_1) \wedge Method(c_2)\big) \vee \big(Task(c_1) \wedge Task(c_2)\big) \vee \big(Other(c_1) \wedge Other(c_2)\big) \quad (16)$$

Actually, keyword-based rules are the basic rules to recognize concepts, and containing-based and apposition rules help to detect some missing concepts (e.g. conditional

random fields) and correct some wrong judgments, so that the concept extraction is more complete.

Transitivity Rules
Actually, some useful concept pairs may fail to be extracted, so we apply transitivity in relation inference. It is supposed that if c_1 and c_2 have a relation, while c_2 and c_3 have a relation, and then concept c_1 and c_3 potentially have a relation. However, the precondition is that they are in the same category. Relations between *Task* and *Method* are not assumed to have transitivity. The rules are represented below:

$$Relation\_m(c_1, c_2) \wedge Relation\_m(c_2, c_3) \Rightarrow Relation\_m(c_1, c_3) \quad (17)$$

$$Relation\_t(c_1, c_2) \wedge Relation\_t(c_2, c_3) \Rightarrow Relation\_t(c_1, c_3) \quad (18)$$

6 Empirical Evaluation

6.1 Evaluation Setup

Dataset and Evaluation Metrics
Our goal is to automatically extract useful knowledge from a set of literatures in a specific domain[1]. Since there exist many different research domains and new research domains emerge rapidly, it is impossible to label training data on each domain in practice. Therefore, experiments will be performed on a cross-domain setting, i.e. the training data and the test data are from different domains. To build the datasets, we collect 200 literature articles from the ACL corpus[2] on 4 domains in the field of natural language processing: "Statistical Machine Translation" (SMT), "Document Summarization" (DS), "Sentiment Analysis and Opinion Mining" (SAOM) and "Reference Resolution" (RR), and each domain contains 50 literature articles. We use articles from ACL corpus because they are well formatted, with pages about 7 to 10, and the text edition is conveniently available on the Internet. In our experiments, only the title, abstract, introduction and related work sections are extracted and used, because these sections have covered most concepts and relations in literature articles. Besides, the other text sections, such as approach and experiments, often contain much noise, like equations, figures and tables. Therefore, we only focus on partial texts of the literatures in this study. We read the texts and manually label the concepts and their relations. In our supervised experiments, we use the labeled data from three domains as training set, and use the labeled data from the remaining one domain as test set. Therefore, four-fold validation are conducted. We calculate the Precision (P), Recall (R) and F-measure (F) values to measure the performance of extractions.

Data Preprocessing
Considering that a concept is usually a noun phrase, we first use a noun phrase (NP) chunking tool - the StanfordNLP toolkit[3] to get all NPs from literature texts.

[1] The size of the literature set in a domain can range from several to thousands.
[2] `http://www.acl.org/` The datasets used will be published on our website.
[3] `http://nlp.stanford.edu/`

The initial NP set contains many useless phrases, such as "we", "this paper", "future work", etc. So we build a simple filter to filter out some NPs by using several linguistic rules, and also collect a "stop words" list to abandon some useless NPs or some useless words in NPs, such as "some", "many", "efficient", "general", etc. In addition, too long or too short NPs are also excluded.

Baselines

To verify the performance of our proposed joint model, we develop three baseline methods (CRF model, SVM classifier, and C-Value method) for concept extraction and a baseline method (SVM classifier) for relation extraction.

The CRF model [16] can be used to extract academic concepts in literatures. To make training data for the CRF model, we search the labeled concepts in the literatures and mark the occurrences of the concepts. If one concept occurs in a literature, each word covered by the concept phrase will be labeled with relevant tags (T, M). The other words are marked with irrelevant tag (O). When two concept phrases are overlapping, such as "machine translation" and "statistical machine translation", we consider the longer one as the complete phrase. The features used in the CRF model include the current word, words around the current word, part of speech and keyword-based features. The lists of keywords are the same as they are for MLN. Besides, to guarantee the fairness of the comparison, NP features are also used in CRF, including a word's position in an NP (i.e. outside, beginning, middle and ending of an NP).

Support Vector Machine [4] is another popular method for information extraction, classifying NPs into three categories. The features used for SVM include prefix and suffix information of NP, and keyword-based features. The keyword-based features also include keywords inside and around concepts, which are used in MLN.

C-Value [10] is an unsupervised algorithm for multi-word terms recognition, and here we refer to concept phrases. It extracts phrases according to the frequency of occurrence of phrases, and enhances the common statistical measure of frequency by using linguistic information and combining statistical features of the candidate NPs.

For concept relation extraction, we use SVM as a baseline. Classifications are conducted on candidate concept pairs, which are combinations of two adjacent concepts. The candidate concept pairs are classified into two categories, having relations or not. The features for classification include phrases' length and position information, relation related keywords and the keywords' position information. Phrases' lengths are calculated by ignoring brackets and the context inside brackets. The position information includes positions in a sentence, the orders of phrase pairs, and the distances between phrases in pairs. Relation related keywords are also collected manually, including phrases like "based on", "enhancement", "developed on", etc.

Alchemy

We utilize the Alchemy system[4] in our experiments. Alchemy is an open-source package providing a series of algorithms for statistical relational learning and probabilistic logic inference, based on the Markov logic representation.

[4] http://alchemy.cs.washington.edu/

6.2 Evaluation Results

Results of Concept Extraction

The number of extracted academic concepts and the performance values of the different methods on four domains are shown in Table 1. The performance values of *Method* extraction, *Task* extraction, and the overall extraction are calculated separately. As C-Value is an algorithm for recognition of useful phrases, and it cannot distinguish the concept types, so only the overall extraction performance is measured. The average values of F-measure are calculated based on F-measures on four domains.

We can see that the overall performance values of our proposed joint model (MLN) are much better than the baselines. Compared to the baseline methods that can only extract concepts independently, the MLN model infers concepts and relations jointly, and it can take into consideration the joint information in a domain. In addition, the rules make it possible to supplement some missing concepts and remove some incorrect concepts. So the extraction results are more accurate and complete.

Table 1. Comparison Results of Concept Extraction

Test Domain		Task Concept			Method Concept			Task + Method			
		MLN	CRF	SVM	MLN	CRF	SVM	MLN	CRF	SVM	CValue
SMT	No.	48	48	84	275	283	293	323	331	377	324
	P	0.875	0.542	0.529	0.920	0.618	0.870	0.913	0.607	0.796	0.574
	R	0.824	0.510	0.870	0.907	0.626	0.916	0.894	0.609	0.909	0.564
	F	**0.849**	0.526	0.658	**0.913**	0.622	0.892	**0.904**	0.608	0.849	0.569
DS	No.	96	107	269	172	181	183	268	288	452	373
	P	0.958	0.701	0.390	0.936	0.586	0.929	0.944	0.628	0.608	0.413
	R	0.836	0.682	0.955	0.885	0.582	0.934	0.866	0.620	0.942	0.527
	F	**0.893**	0.691	0.554	0.910	0.584	**0.932**	**0.904**	0.624	0.739	0.463
SAOM	No.	185	200	534	323	353	351	508	553	885	455
	P	0.789	0.750	0.322	0.944	0.572	0.895	0.888	0.637	0.549	0.673
	R	0.741	0.761	0.873	0.892	0.591	0.918	0.837	0.653	0.902	0.568
	F	**0.764**	0.756	0.471	**0.917**	0.581	0.906	**0.862**	0.645	0.683	0.616
RR	No.	95	111	281	177	196	195	272	307	476	297
	P	0.958	0.748	0.320	0.972	0.607	0.903	0.967	0.658	0.559	0.731
	R	0.858	0.783	0.849	0.901	0.623	0.921	0.886	0.680	0.896	0.731
	F	**0.905**	0.765	0.465	**0.935**	0.615	0.912	**0.924**	0.669	0.688	0.731
Average	P	0.895	0.685	0.390	0.943	0.596	0.899	0.928	0.633	0.628	0.598
	R	0.815	0.684	0.887	0.896	0.606	0.922	0.871	0.641	0.912	0.597
	F	**0.853**	0.685	0.537	**0.919**	0.601	0.911	**0.899**	0.637	0.740	0.595

We can also find that the extraction of *Methods* is generally more effective than the extraction of *Task*s. Considering the characteristics of academic literatures, they are usually discussing several methods for one task, so the number of *Method*s is certainly larger than that of *Task*s, which can be confirmed from the numbers of extracted concepts in Table 1. With more ground atoms of *Method* in training data, the MLN model can learn a better model and performs more effectively for *Method* extraction.

Finally, we can conclude that our joint model helps to extract concepts in a more comprehensive way by taking into account global information and the relations between concepts. This advantage will be more notable in relation extraction.

Results of Relation Extraction

In our MLN model, concepts and relations are extracted together. While in the baseline, concepts are firstly extracted by the baseline model with the best performance (i.e. the SVM model), and then the SVM classifier is used for relation classification based on the extracted concept pairs. The results are shown in Table 2.

The performance of MLN is much better than SVM. Making full use of joint information, the joint model helps to improve the performance of knowledge extraction.

Although the SVM classifier returns more relation pairs, the recall value it gets is lower than that of MLN. This shows that the SVM classifier brings more incorrect results and its ability of discrimination between positive and negative cases in relation extraction is relatively poor.

Table 2. Results of Relation Extraction

	SMT				**DS**			
	No.	**P**	**R**	**F**	**No.**	**P**	**R**	**F**
MLN	469	0.85	0.82	**0.84**	277	0.90	0.74	**0.81**
SVM	668	0.52	0.71	0.60	368	0.67	0.73	0.70
	SAOM				**RR**			
	No.	**P**	**R**	**F**	**No.**	**P**	**R**	**F**
MLN	745	0.92	0.76	**0.83**	344	0.94	0.86	**0.90**
SVM	687	0.69	0.53	0.60	440	0.53	0.62	0.57
	Average performance in four domains							
MLN	**P**: **90.2%**; **R**: **79.6%**; **F**: **84.6%**.							
SVM	**P**: 60.1%; **R**: 64.7%; **F**: 62.3%.							

7 Knowledge Graph Generation

7.1 Concept Importance Assessment

Before generating knowledge graphs, we assess the importance of different concepts in a specific domain, which is the basis of concept cloud graph generation. Importance assessment is based on the frequency a concept occurs in the literatures and the

importance of other concepts that have relations with this concept. We have two assumptions:

- The more frequently a concept appears, the more important it is.
- The more important the other concepts related to the current concept are, the more important the current concept is.

In addition, the position where a concept occurs is also considered into importance assessment. For instance, a concept occurs in title will get a larger importance weight than the ones in other sections. The position-based weight of a concept is multiplied with the frequency it occurs in the corresponding section. In this study, each concept's prior importance score is given by its frequency, and the frequency of a concept occurs in title is multiplied by a weight (we assign 1.2 in experiments).

In the experiments, the personalized PageRank algorithm [12] is applied for concept importance assessment. The personalized PageRank algorithm can take into account both the prior scores and the concept relations, and the importance score of a concept is iteratively calculated until convergence.

7.2 Concept Cloud Graph Generation

In order to help researchers have a clearer and more comprehensive understanding of the concepts, concept cloud graph is generated according to the importance of concepts. Given the numerical importance scores of academic concepts, concept-clouds are produced by the "Wordle" platform[5].

As a case, the concept-cloud graph for the "DS" domain is shown in Figures 2. Due to the limited space, the graph only contains a portion of the concepts. After taking a look at the graph, we can know the most important concepts in the "DS" domain clearly and quickly. It is obvious that the most notable concept is "multi-document summarization", while "document summarization" and "single-document summarization" are relatively less notable. This implies that more researches focus on multi-document summarization nowadays. The concept "topic detection and tracking" is also notable, as it has many associations with "Document Summarization". In addition, concepts with larger size are mostly *Task* concepts, indicating that *Task* concepts occur more frequently in literatures.

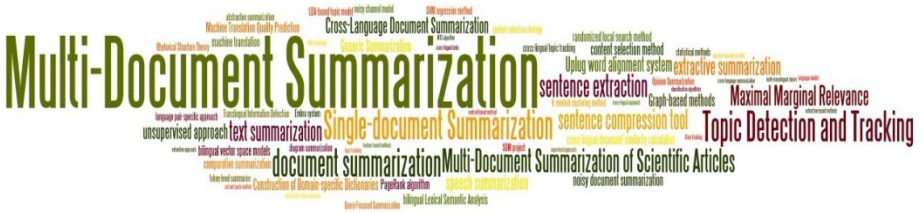

Fig. 2. Concept Cloud Graph on "DS" Domain

[5] http://www.wordle.net/advanceds

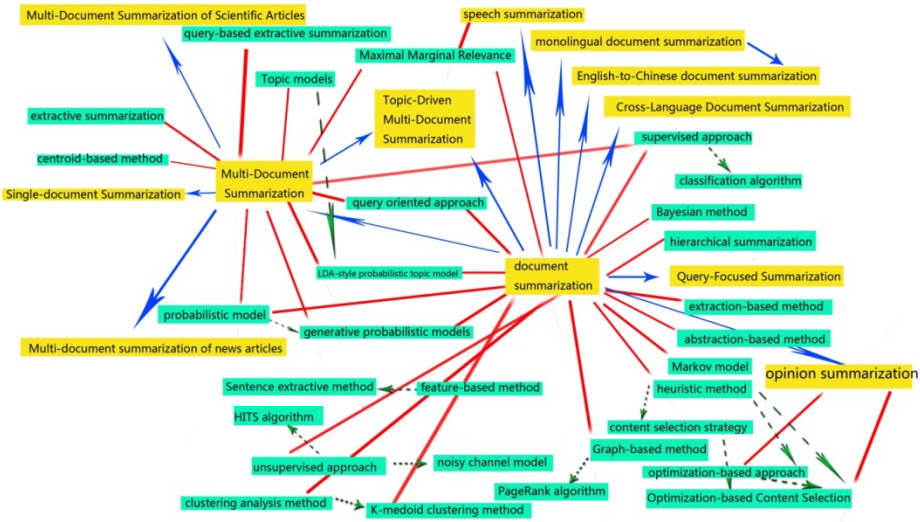

Fig. 3. Part of Relations Graph on "DS" Domain

(Yellow boxes denote *Task*s, and green boxes represent *Method* concepts. Red arrows denote *Task-Method* relations, blue arrows denote *Task-Task* relations, and green arrows denote *Method-Method* relationships.)

7.3 Concept Relation Graph Generation

In order to present the extracted concept relations in academic literatures vividly, a concept relation graph is built on a research domain. Taking the "DS" domain as an example, the concept relation graph is shown in Figure 3. Again because of space limitation, only a portion of concept relations are shown. From the graph, we can learn that the "document summarization" *Task* evolves into a few *Tasks*, such as "single-document summarization", "multi-document summarization", "monolingual document summarization", "query-focused summarization" and so on. Many *Methods* applied to these *Tasks* are shown directly and vividly, and relations between *Method*s are also shown visually.

8 Conclusions and Future Work

In this paper, we propose a novel AKMiner system to extract academic concepts and relations between concepts from academic literatures in a research domain. The extracted results are visually presented to users via concept-cloud graph and concept relation graph. In future work, we will further classify relations into more specific categories, and even relations among three or more concepts will be investigated. We will also explore to recommend *Method*s for some *Task*s through analysis of the network of concepts and their relations. This will bring much inspiration to scientific research work.

Acknowledgments. The work was supported by NSFC (61170166) and National High-Tech R&D Program (2012AA011101).

References

1. Abu-Jbara, A., Radev, D.: Coherent Citation-Based Summarization of Scientific Papers. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pp. 500–509 (2011)
2. Agarwal, N., Gvr, K.: SciSumm: A Multi-Document Summarization System for Scientific Articles. In: Proceedings of the ACL-HLT 2011 System Demonstrations, pp. 115–120 (2011)
3. Church, K.W., Hanks, P.: Word association norms, mutual information and lexicography. In: ACL 1989, pp. 76–83 (1989)
4. Cortes, C., Vapnik, V.: Support-vector Networks. Machine Learning 20(3), 273–297 (1995)
5. Daille, B.: Combined approach for terminology extraction: lexical statistics and linguistic filtering. Technical Report (1995)
6. Dunne, C., Shneiderman, B., Gove, R., Klavans, J., Dorr, B.: Rapid Understanding of Scientific Paper Collections: Integrating Statistics, Text Analytics, and Visualization. University of Maryland, Human-Computer Interaction Lab. Tech. Report HCIL-2011 (2011)
7. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics 19(1), 61–74 (1994)
8. Earl, L.L.: Experiments in automatic extracting and indexing. Information Storage and Retrieval 6(X), 273–288 (1970)
9. EL-Arini, K., Guestrin, C.: Beyond Keyword Search: Discovering Relevant Scientific Literature. In: Proceedings of the 17th SIGKDD (2011)
10. Frantzi, K., Ananiadou, S., Mima, H.: Automatic recognition of multi-word terms: the C-value / NC-value method. International Journal of Digital Library 3, 115–130 (2000)
11. Han, H., Giles, C., Manavoglu, E., Zha, H., Zhang, Z., Fox, E.: Automatic Document Meta-data Extraction using Support Vector Machines. In: Proceedings of Joint Conference on Digital Libraries (2003)
12. Haveliwala, T.H.: Topic-sensitive pagerank. In: WWW 2002, pp. 517–526. ACM, New York (2002)
13. Isaac, Councill, G., Giles, C.L., Kan, M.Y.: ParsCit: An open-source CRF reference string parsing package. In: Proceedings of the Language Resources and Evaluation Conference (LREC 2008), Marrakesh, Morrocco (May 2008)
14. Jiang, X., Hu, Y., Li, H.: A ranking approach to keyphrase extraction. In Microsoft Research Technical Report (2009)
15. Justeson, J., Katz, S.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering 1, 9–27 (1995)
16. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of ICML, pp. 282–289 (2001)
17. Li, N., Zhu, L., Mitra, P., Mueller, K., Poweleit, E.: oreChem ChemXSeer: a semantic digital library for chemistry. In: JCDL (2010)
18. Mihalcea, R., Tarau, P.: TextRank: Bringing order into texts. In: Proceedings of EMNLP 2004 (2004)
19. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Libraries (1998)
20. Poon, H., Domingos, P.: Joint inference in information extraction. In: Proceedings of AAAI 2007 (2007)

21. Poon, H., Vanderwende, L.: Joint inference for knowledge extraction from biomedical literature. In: Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (2010)
22. Poon, H., Domingos, P.: Joint unsupervised coreference resolution with Markov logic. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (2008)
23. Qazvinian, V., Radev, D.R.: Scientific Paper Summarization Using Citation Summary Networks. In: Proceedings of COLING 2008, vol. 1, pp. 689–696 (2008)
24. Richardson, M., Domingos, P.: Markov Logic Networks. Machine Learing 62(1-2), 107–136
25. Shahaf, D., Guestrin, C., Horvitz, E.: Metro Maps of Science. In: Proceedings of the 18th ACM SIGKDD (2012)
26. Singla, P., Domingos, P.: Entity resolution with markov logic. In: Proceedings of ICDM 2006 (2006)
27. Singla, P., Kautz, H., Luo, J.: Discovery of social relationships in consumer photo collections using Markov Logic. In: Workshops of CVPRW 2008 (2008)
28. S.K., Kan, M.: Scholarly paper recommendation via user's recent research interests. In: JCDL (2010)
29. Kondo, T., Nanba, H., Takezawa, T., Okumura, M.: Technical Trend Analysis by Analyzing Research Papers' Titles. In: Vetulani, Z. (ed.) LTC 2009. LNCS, vol. 6562, pp. 512–521. Springer, Heidelberg (2011)
30. Wan, X.J., Xiao, J.G.: Single document keyphrase extraction using neighborhood knowledge. In: Proceedings of AAAI 2008 (2008)
31. Yeloglu, O., Milios, E., Zincir-Heywood, N.: Multi-document Summarization of Scientific Corpora. In: SAC (2011)
32. Zhu, J., Nie, Z., Liu, X., Zhang, B.: StatSnowball: a statistical approach to extracting entity relationships. In: Proceedings of 18th WWW Conference (2009)