IExM: Information Extraction System for Movies*

Peng-Yu Chen
National Tsing Hua University
Hsinchu, Taiwan
pengyu@nlplab.cc

Yi-Hui Lee
National Taiwan Normal University
Taipei, Taiwan
amy030619@gmail.com

Yueh-Han Wu
National Tsing Hua University
Hsinchu, Taiwan
dgrey1116@gmail.com

Wei-Yun Ma
Institute of Information Science, Academia Sinica
Taipei, Taiwan
ma@iis.sinica.edu.tw

ABSTRACT

In this demonstration, we present Information Extraction System for Movies(IExM), which helps extract relation instances from unlabeled movie articles. We have designed a new distant-supervised learning algorithm: Improved Pattern Ranking Algorithm(IPRA) to extract relation instances from unlabeled articles, which iteratively generates new patterns starting from a limited set of seed instances, and extracts new instances using high-ranking pattern in a precise and effective way. IPRA also has a special estimation for the newly generated patterns based on the quality estimation of the instances that generate the patterns and ranks patterns' quality based on various factors.

Keywords

information extraction; pattern generation; pattern ranking; Wikipedia; E-HowNet; bootstrapping; distant supervision; semi-supervised learning; relation extraction; infobox

1. INTRODUCTION

Wikipedia provides infobox to help users gain the information they want conveniently, however, there are still a lot of wiki pages with incomplete infobox. Since manually constructing the infobox is too expensive, we develop a system to automatically extract the structured information from unstructured text data. Combining wiki pages and web resources, we present Information Extraction System for Movies(IExM), which helps extract relation instances from unlabeled movie articles. To complete the system, we need to use some technologies discussed below.

Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine readable documents. To overcome the problem of the lack of training data, Mintz *et al.* [1] proposed Distant-supervised learning algorithm (DSLA) to generate training data from Freebase. Another strategy is through bootstrap learning to extract more patterns from seeds or instances in an iterative fashion. However bootstrapping often suffer from semantic drift problem [2]. To address this issue, two approaches are common in use. NELL [3, 6, 7] proposes the coupled training method by building a large amount of coupled relations and setting the mutually exclusive constraints between these coupled relations. Sun and Grishman [4] designed a pattern ranking algorithm with pattern clustering strategy to prevent semantic drift. However, their method does not update patterns' quality and also fail to consider the quality of the instances that generate these patterns.

To address the above considerations, we propose Improved Pattern Ranking Algorithm (IPRA) for IE tasks, which estimates patterns' quality according to various factors, including the patterns' occurrence and coverage of application, and the quality estimation of the instances which are actually extracted by these patterns. The experimental results show that as more patterns are generated and ranked, the coverage and precision of extracted instances can be gradually improved and then achieve a high performance in the end.

The related work are introduced in Section 2. In Section 3 and 4, the details of our system and IPRA model are introduced. The performance evaluation on the proposed methods and related works is reported in Section 5.

2. RELATED WORK

At least four learning paradigms of information extraction have been presented for the task of extracting relation from text. First paradigm is to manually design patterns for a rule-based approach, which is born with the defects that it takes much human efforts and lacks flexibility. The second paradigm is through a supervised learning procedure: to build a large-scale, machine learning classifier to judge if a given entity pair has a certain relation. Since it requires a large amount of labeled data, a lot of manual efforts are also needed.

Another common paradigm is through bootstrapping method for semi-supervised learning [2, 5]: to begin with a limited number of labeled instances in context and much more unlabeled documents in a specific domain, and extract patterns as extractors. The extracted instances are used with a large corpus to generate a new set of patterns. The generated patterns are then used to extract more instances. Each time the process involving a stage of "instances gen-

*The work described in this paper was done as part of (and partially supported by) 2016 Summer Internship Program at the Academia Sinica, Institute of Information Science, Taiwan.

WWW 2017 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3054729

erate patterns" and a stage of "patterns extract instances" is called an iteration. Brin *et al.* [2] present a technique which exploits the duality between sets of patterns and relations to grow the target relation starting from a small sample, and test the extract relation (author,title) pairs from the World Wide Web. However, it often causes "semantic drift" problem after many iterations [2, 5]. e.g., For extracting the 'country' from the movie articles, 'the United States' is the target instance of the pattern 'film in', but the pattern generated from 'the United States' could be 'live in'. New instances generated from the pattern 'live in' may be 'apartment building'. In this situation, we get the instance 'apartment building', but it's not an instance of 'country'. The semantic meaning of the target instance deviates.

For solving the semantic drift problem efficiently, there are two methods which are commonly used. One is coupled training, the other is pattern ranking. NELL [3, 6, 7] makes use of coupled training method to build a large number of coupled relations. With an initial ontology defining categories and about a dozen labeled training examples for each category and relation, NELL extracts candidate instances by patterns and evaluates the quality of the candidate instances by the number of promoted patterns that they co-occur with [3]. They also set mutual exclusive constraints between these coupled relations. NELL has been learning to read the web 24 hours/day since January 2010, and so far has acquired a knowledge base with over 80 million confidence-weighted beliefs (e.g., servedWith(tea, biscuits)). On the other hand, Sun and Grishman [4] design the pattern ranking algorithm. A pattern ranking algorithm with pattern clustering strategy is presented to prevent semantic drift. While pattern clustering strategy does bring benefits, their framework only estimates patterns' quality based on the instances (and their clusters) that these patterns can match, and accept a certain number of top ranked patterns. Their method does not update patterns'qualities based on the instances that the top ranked patterns actually generated and also fail to consider the quality of the instances that generate the patterns.

Another similar paradigm is Distant-supervised learning algorithm (DSLA). It is similar with semi-supervision, and the only difference is the seeds/instances are usually certain target objects, such as attributes, instead of labeled instances in context.

For the pattern design, context pattern [2, 5] and syntactic pattern are common in use. Our work investigates syntactic patterns and mixed context patterns, combining three different semantic units in the pattern design: word, part-of-speech tag and word sense.

3. SYSTEM DESCRIPTION

In this section, we will describe how our system, IExM, works based on IPRA model. The system is presented in the form of a web application[1], where you can input a movie title and the attribute you wish to know. Once you click the 'search' button, the system will collect articles from Wikipedia and other websites related to the movie, extracting the target attribute and list the result. The system will also highlight the target attribute and the pattern in the result. Figure 1 shows the system screenshot, which we will demonstrate at the conference. You can also find the screencast in the same website.

The system architecture (as illustrated in Fig. 2) consists of three main components, the pretrained model using Wikipedia data with IPRA(we will describe this model in detail in next section), the Wikipedia database, and a user interface. Once a user types the movie name with an attribute which he wants to search, our system first collects some related articles from the web with the movie

[1]http://learn.iis.sinica.edu.tw/IExM

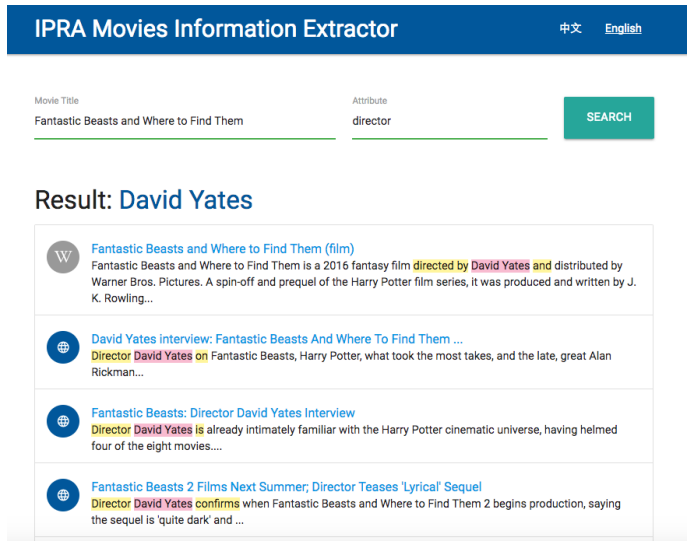


Figure 1: The screenshot of the system

name as keyword, then for every article, we try to find a pattern in our model that can match this article. Based on all the matched articles and the information in Wikipedia database, the system chooses an appropriate answer and shows the result to user.

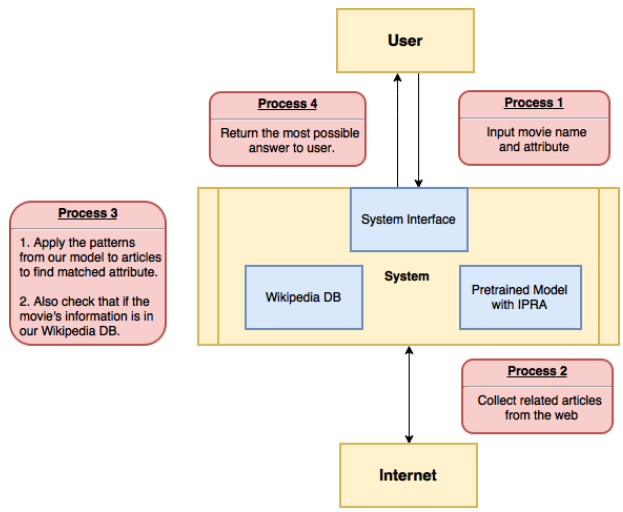


Figure 2: System architecture

We are still extending the data domain of our system. In the future, our system will have a variety of knowledge in different domains in addition to movie information, and this would also be useful on many NLP tasks.

4. MODEL

We developed an information extraction model with an Improved Pattern Ranking Algorithm, named IPRA. IPRA can extract attributes from an article of a specific theme. The attribute values extracted are called **instances**, and the initial manually chosen instances are **seeds**. To simplify the problem, we focus on Chinese Wikipedia articles and choose movies and TV series as our domain categories.

4.1 Pattern Design

We have two main types of pattern: the context pattern and the syntactic pattern. A context pattern consists of the context information of the target attribute, and a syntactic pattern focuses on the sentence structure in which the target attribute occurs.

4.1.1 Context Pattern

Suppose we have a sentence:

《斷背山》由台灣導演李安執導
English: Brokeback Mountain is directed by Ang Lee, a Taiwanese director.

After Chinese word segmentation and part-of-speech tagging, we get the result shown in Figure 3.

POS	parenthesis	Nc	parenthesis	P	Nc	Na	Nb	VC
word	《	斷背山 (Brokeback Mountain)	》	由 (by)	台灣 (Taiwan)	導演 (director)	李安 (Ang Lee)	執導 (directed)

Figure 3: Word segmentation and POS tagging

Our target attribute value, the director of the movie, is 李安(Ang Lee). Now we need to look at the context of the word and transform that into a pattern. Assuming the window size of the context is 1, and the target attribute is denoted as (.+?) in regular expression, we have four types of context pattern described below:

1. **Word**
   Only looking at the left adjacent word and the right adjacent word of the target.
   pattern: <導演>(.+?)<執導>
   (<director>(.+?)<directed>)

2. **POS**
   A POS(part-of-speech tagging) pattern can extract more attributes than a word pattern can, although this may also decrease the precision.
   pattern: <Na>(.+?)<VC>

3. **E-HowNet word sense**
   The term 'word sense' means a general representation of a word. The definition of sense we adopt is based on E-HowNet(Extended-HowNet)[2].
   pattern: <human|人.1>(.+?)<undertake|擔任.1>

4. **Mixed**
   Combining the three types above, we create a mixed pattern type. Assuming the context window size is 2, we have 4 positions for 3 kinds of pattern type. So the number of different patterns is $3^4 = 81$

   - word word (.+?) word wod
   - word word (.+?) word pos
   - ...
   - sense sense (.+?) sense pos
   - sense sense (.+?) sense sense

[2]http://ehownet.iis.sinica.edu.tw/index.php

4.1.2 Syntactic Pattern

Given the sentence '《斷背山》由台灣導演李安執導', we can obtain a syntactic tree structure by using CKIP Chinese Parser. Leaf nodes being the segmented words, each internal node has two values, the semantic role and the part-of-speech tagging of the subtree. Leveraging on this tool, we create two kinds of syntactic patterns.

1. **Parse tree path**
   Taking the semantic roles along the path from root node to the node of our target attribute, we create a pattern that can carry some syntactic information of the sentence.

2. **Parse tree path with head word**
   Only looking at the tree path may be too ambiguous. A path consisting of some semantic roles might have nothing to do with the attribute itself. For this reason, we add another factor, head word, into the pattern, In a syntactic tree, head word can usually capture the key intention of the sentence.

4.2 Improved Pattern Ranking Algorithm

With a sample of seeds, IPRA iteratively generates patterns and extracts attributes from chosen Wikipedia articles. Figure 4 shows the framework of IPRA. A single iteration of the process consists of four stages, which is described below.

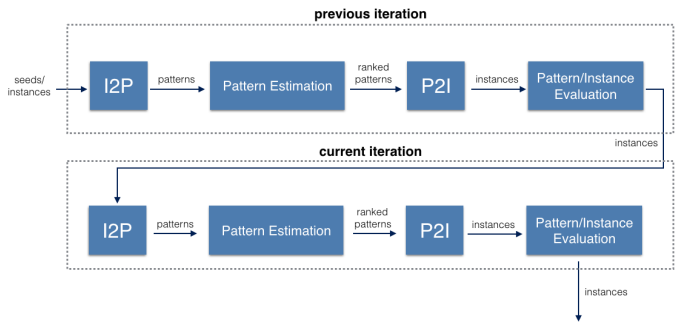


Figure 4: IPRA framework.

4.2.1 Instance-To-Pattern Stage(I2P)

With given attribute value in an article, we find the sentences where the attribute occurs, and transform the way the attribute is mentioned into our pattern format.

4.2.2 Pattern-To-Instance Stage(P2I)

The patterns generated through instance-to-pattern stage can then be used to extract new instances. Before starting the extraction process, we use our pattern ranking algorithm to rank the new generated patterns and all the previously generated patterns. The pattern with the highest rank would be used to extract attributes first, and the patterns with lower rank would only be used on the articles whose attribute values are not yet extracted. To observe the performance of each iteration, before going back to the instance-to-pattern stage, we evaluate all the instances using the infobox of each article and calculate the precision and coverage, shown in Figure 5.

4.2.3 Pattern/Instance Evaluation

In pattern-to-instance stage, how should we decide which pattern to apply first if there were several patterns to choose from? We propose to estimate patterns' quality according to various factors,

including the patterns' occurrence, the coverage and the quality estimation of the instances which are actually extracted by these patterns. Based on these assumptions, we developed a new pattern ranking algorithm to score a pattern before and after it is used in pattern-to-instance stage. In the rest of this subsection, we will introduce how to estimate the quality of pattern after P2I and introduce the pattern estimation before P2I on 4.2.4.

First, we define how to measure the quality of an instance. We call it **precision**. The precision of seed instances are initialized to 1, while the precision of other instances are initialized to 0. Suppose the instance I_i is obtained from k source patterns. At the end of each iteration, we recalculate the precision of each instance using the equation below:

$$Prec(I_i) = \frac{\sum_{j=1}^{k} Conf(P_j)}{k} patternbeforeandafter \quad (1)$$

$Conf(P_j)$ is the **confidence** of pattern P_j, which is described below. To calculate the score of a pattern, we consider three factors:

1. **Term frequency(TF)**: The number of times the pattern being applied in the corpus.

2. **Document frequency(DF)**: The number of documents in which the pattern is applied.

3. **Confidence(Conf)**: A value indicating whether the pattern is generated from a good instance. Suppose the confidence of a pattern P_i is denoted as $Conf(P_i)$, and the precision of its k source instances I_j is $Prec(I_j)$, the equation is:

$$Conf(P_i) = 1 - \prod_{j=1}^{k}(1 - Prec(I_j)) \quad (2)$$

The equation 2 shows that the confidence of a pattern would remain to be 1 if any of its source instances comes from the seed instances. We normalize the value of each factor to [0, 1] and calculate the score by taking the average of the three values.

$$PatternScore(P_i) = \frac{TF(P_i) + DF(P_i) + Conf(P_i)}{3} \quad (3)$$

We keep track of instance precision and pattern score by maintaining an instance precision table and a pattern ranking table, updating the values in the end of each iteration.

4.2.4 Pattern Estimation

When a new pattern is generated from an instance, we can calculate its confidence, but how can we know the TF and DF before it is applied to extract attributes? We can not calculate the real score. Therefore, we present a special estimation for the newly generated patterns based on the quality estimation of the instances that generate the patterns, and these quality estimation of the instances are related to the source patterns that generate these instances. We define the $InstanceScore$ of an instance to be the weighted average of the scores of all of its source patterns. The weight is the reciprocal of the rank number of the pattern.

$$InstanceScore(I_i) = \frac{\sum_{j=1}^{k}(PatternScore(P_j) \times \frac{1}{rank(P_j)})}{\sum_{j=1}^{k} \frac{1}{rank(P_j)}} \quad (4)$$

Then, we can calculate the $EstimatedPatternScore$ of the newly generated pattern by taking the average of its source instances' $InstanceScore$.

$$EstimatedPatternScore(P_i) = \frac{\sum_{j=1}^{k} InstanceScore(I_j)}{k} \quad (5)$$

The estimated score of the pattern is overwritten by the real score once the pattern finishes pattern-to-instance stage.

5. EXPERIMENT

5.1 Data and Preprocessing

We dump all Chinese articles from Wikipedia as our experimental resource which include both unstructured text and infobox. An infobox is a fixed-format table on Wikipedia page designed to consistently present a summary of some unifying aspect that the articles share. We collect articles list from Wikipedia's categories called movie and TV series and all its subcategories, which contain 4694 movie articles and 5817 TV series articles, as our domain, making use of the information from the infobox such as director, country, and screenwriter to produce initiative seeds and also take the infobox as the golden answer to evaluate the result.

In order to generate part-of-speech and parse tree for the syntactic and context pattern, we use CKIP Chinese Word Segmentation System[3] and CKIP Chinese Parser[4]. These tools have high accuracy in Chinese environment compared to others. Besides, we also use the Extended-HowNet(E-HowNet) to expand our context pattern. E-HowNet is the lexical semantic representation model for natural language understanding. Changing the context pattern from word layer to sense layer enhances the ability to match more instances.

5.2 Result

Each article could contain varying attributes in the infobox. For extracting the specific attribute to evaluate our method, we select three attributes: "director", "country" and "screenwriter". We pick up 4105, 3895, 710 articles from 4694 movie articles and 5817 TV series articles which have attribute "director", "country" and "screenwriter" in the infobox respectively. For held-out evaluation experiments, we randomly pick up 20 articles as our seeds to see if our method could obtain the specific attribute from the rest articles.Our method gets better F1-score when we use pos-based context patterns and set the window size to 2. And word-based context patterns gets better precision. The result is shown in Table 1 and 2.

Table 1: Result of Different Attributes

	Precision	Recall	F1-Score
Director	86.1%	63.8%	73.3%
Country	80.1%	69.4%	74.4%
Screenwriter	99.0%	55.6%	71.2%

5.2.1 Pattern Ranking Algorithm

We choose the best pattern type: **pos pos target pos pos** to compare the performance of two baselines and our IPRA algorithm. The two baselines are **Voting**: collecting the instances extracted by new patterns in one article, and then decide the instance

[3]http://ckipsvr.iis.sinica.edu.tw/
[4]http://parser.iis.sinica.edu.tw/

Table 2: word/pos/sense/mixed(top4) Context Patterns

Pattern Type	Precision	Recall	F1-Score
word word target word word	**90.2%**	55.3%	68.6%
pos pos target pos pos	86.1%	**63.8%**	**73.3%**
sense sense target sense sense	89.7%	56.0%	68.9%
pos pos target pos word	85.7%	63.2%	72.7%
pos pos target pos sense	85.7%	63.4%	72.7%
word pos target pos pos	87.8%	61.9%	72.6%
word pos target pos word	88.0%	61.6%	72.5%

with highest votes, and **Confidence-Based Pattern Ranking Algorithm (PRA)**: considering only the confidence of pattern to estimate the patterns' quality, that is, using only equation 1 and equation 2. Figure 5 obviously shows that our approach IPRA(f1-score: 0.733) performs better than voting method(f1-score:0.670) and confidence-based PRA(f1-score: 0.680), which shows the reasonable assumption that the dynamic evaluation for pattern enhances the performance.

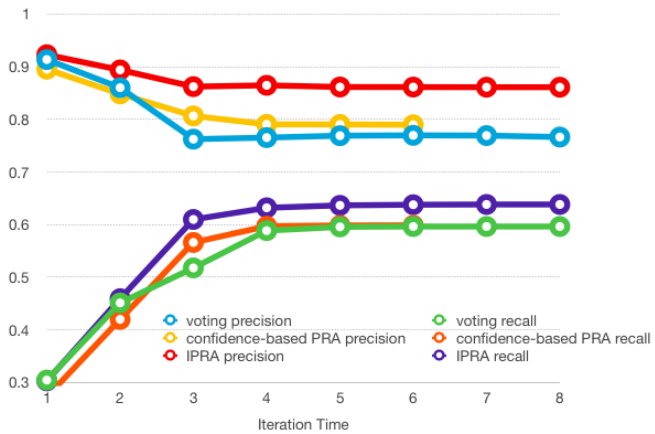


Figure 5: Algorithms' performance comparison

5.2.2 *Missing Information in the Infobox*

We manually evaluate about 600 articles which do not have the attribute "director" in the infobox. By human evaluation, the attribute "director" is mentioned in the 179 articles. With our method, we find out the director from the 101 articles among the 179. Although there are 78 articles missed or have wrong answer, we can still get a precision of 77% and recall of 56% which shows in Table 3. The experiment prove that our method has the potential to extract information from the context correctly, and it can be applied to a variety of tasks including expanding Wikipedia and other corpus.

Table 3: 589 articles which miss 'director' attribute

	Found	Not found
Director appears in context	101	78
Director doesn't appear in context	31	379
Precision: 77% , Recall: 56% , F1-Score: 65%		

6. REFERENCES

[1] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. *Distant supervision for relation extraction without labeled data*. In AFNLP.

[2] S. Brin. *Extracting patterns and relations from the world wide web*. In WebDB Workshop at 6th Intl. Conf. on Extending Database Technology, 1998.

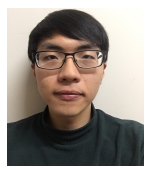
[3] A. Carlson, J. Betteridge, R.C. Wang, E.R. Hruschka Jr. and T.M. Mitchell. *Coupled Semi-Supervised Learning for Information Extraction*. In WSDM, 2010.

[4] Ang Sun and Ralph Grishman. 2010. *Semi-supervised Semantic Pattern Discovery with Guidance from Unsupervised Pattern Clusters*. In COLING.

[5] Ellen Riloff and Rosie Jones. *Learning dictionaries for information extraction by multi-level bootstrapping*. In Proc. of AAAI, 1999.

[6] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr. and T.M. Mitchell. *Toward an Architecture for Never-Ending Language Learning*. In Proceedings of the Conference on Artificial Intelligence (AAAI), 2010.

[7] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, J. Welling. *Never-Ending Learning*. In Proceedings of the Conference on Artificial Intelligence (AAAI), 2015.



Peng-Yu Chen received the B.S. degree of Computer Science from National Tsing Hua University, Hsin-Chu, Taiwan in 2014. He is currently working towards the Graduate degrees of Computer Science, National Tsing Hua University. His research interests include natural language processing, data mining, and web development.

Yi-Hui Lee received the B.S. degree of Computer Science from National Taiwan Normal University, Taipei, Taiwan, in 2015. She is currently working towards the Graduate degrees of Computer Science at National Taiwan Normal University. Her main research interests include data mining, information retrieval, machine learning, natural language processing.

Yueh-Han Wu is currently pursuing B.S. in Computer Science at National Tsing Hua University, Taiwan. He spends lots of time doing web development and natural language processing.

Wei-Yun Ma received his M.S. degree and Ph.D. degree from Columbia University in 2008 and 2014, respectively. Currently, he is an assistant research fellow of the Institute of Information Science, Academia Sinica, focusing on sematic analysis of social media and machine reading.

His research interests include natural language processing, natural language understanding, machine learning, deep learning, machine translation and knowledge representation.