Proceedings of
Open Knowledge Base and Question Answering
Workshop at SIGIR2017

11 August, 2017

Tokyo, Japan

Organization

Organizing Committee

Key-Sun Choi (KAIST, Korea)
Teruko Mitamura (Carnegie Mellon University, USA)
Piek Vossen (Vrije Universiteit Amsterdam, Netherlands)
Jin-Dong Kim (Database Center for Life Science, Japan)
Axel-Cyrille (Ngonga Ngomo, Universität Leipzig, Germany)

Programme Committee

André Freitas (University of Passau, Germany)
Axel-Cyrille Ngonga Ngomo (University of Leipzig, Germany)
Christina Unger (Universität Bielefeld, Germany)
Eun-Kyung Kim (KAIST, Korea)
Jin-Dong Kim (Database Center for Life Science (DBCLS), Japan)
Key-Sun Choi (KAIST, Korea)
Piek Vossen (Vrije Universiteit Amsterdam, Netherlands)
Pum-mo Ryu (Busan University of Foreign Studies, Korea)
Ricardo Usbeck (University of Leipzig, Germany)
Jun Araki (Carnegie Mellon University, USA)
Peter Clark (Allen Institute for AI, USA)
Eduard Hovy (Carnegie Mellon University, USA)
Madoka Ishioroshi (National Institute of Informatics, Japan)
Hiroshi Kanayama (IBM Research - Tokyo, IBM Japan)
Bernardo Magnini (Fondazione Bruno Kessler, Italy)
Tatsunori Mori (Yokohama National University, Japan)
Yuta Nakashima (Osaka University, Japan)
Eric Nyberg (Carnegie Mellon University, USA)
Anselmo Peñas (UNED, Spain)
John M. Prager (IBM T.J. Watson Research Center, USA)
Kotaro Sakamoto (Yokohama National University, Japan)
Hideyuki Shibuki (Yokohama National University, Japan)
Hideki Shima (Duolingo, Inc., USA)
Koichi Takeda (IBM Research - Tokyo, IBM Japan)
Chuan-Jie Lin (National Taiwan Ocean University, Taiwan)
Di Wang (Carnegie Mellon University, USA)
Luke Zettlemoyer (University of Washington, USA)

Introduction

The huge and rapidly increasing amount of structured and unstructured data available on the Web makes it both possible and necessary to support users in finding relevant information. The trend moves more and more towards smart knowledge services that are able to find information, aggregate them, draw inferences, and present succinct answers without requiring the user to wade through a large number of documents. The novel avenues made possible by knowledge services are numerous and diverse, including ubiquitous information access (from smartphones, tablets, smart watches, etc.), barrier-free access to data (especially for the blind and disabled) and knowledge discovery.

Over the last years, several challenges and calls for research projects have pointed out the dire need for pushing natural language interfaces. In this context, the importance of Semantic Web data as a premier knowledge source is rapidly increasing. But we are still far from having accurate natural language interfaces that allow handling complex information needs in a user-centric and highly performant manner. The development of such interfaces requires the collaboration of a range of different fields, including natural language processing, information extraction, knowledge base construction and population, reasoning, and question answering.

The main goal of this workshop is to join forces in the collaborative development of open frameworks for knowledge extraction and question answering, to share standards, and to foster the creation of an ecosystem of tools and benchmarks. The workshop will therefore not only comprise short and long paper presentations but also a hands-on session on already existing frameworks, standards, and benchmarking campaigns, as well as a social meet-up.

The program includes 6 oral presentations on original research papers, and 3 demo presentations on working systems. An invited talk will deliever an in-depth introduction about QA for university entrance exam. In the end of the day, a panel discussion session will be organized jointly with the KG4IR workshop.

We wish to express our gratitude to all the authors for the time and energy they invested in their research and for their choice of OKBQA2017 as the venue to present their work. We are indebted to all members of the programme committee for their detailed inspection of all submitted work and their valuable comments. Additional thanks go to the panelists and the invited speaker for accepting our invitation and delivering inspiring talks.

We hope all the audience to fully enjoy the workshop.

OKBQA 2017 Organizers

Program of December 12th, Thursday

08:50 - 09:00	Opening
09:00 - 10:30	Session I
09:30 10:00 10:30	Controlling Expressiveness of Question Interpretation with a Constrained Tree Transducer Induction *Pascual Martínez-Gómez and Yusuke Miyao* TaxoPhrase: Exploring Knowledge Base via Joint Learning of Taxonomy and Topical Phrases *Weijing Huang, Wei Chen, Tengjiao Wang and Shibo Tao* Multilingualization of Question Answering Using Universal Dependencies *Hiroshi Kanayama and Koichi Takeda*
10:30 - 10:50	Coffee break
10:50 - 12:20	Session II
11:30 12:00 12:20	Challenges in QA for University Entrance Exam at NTCIR QA-Lab: From Multiple-Choice to Essay Questions $^{\underline{\textit{Invited}}}$ *Hideyuki Shibuki (Yokohama National University)* Wikipedia Based Essay Question Answering System for University Entrance Examination *Takaaki Matsumoto, Francesco Ciannella, Fadi Botros, Evan Chan, Cheng-Ta Chung, Keyang Xu, Tian Tian and Teruko Mitamura* Demo introductions (3 presentations, 6 min. each)
12:20 - 14:00	Lunch break
14:00 - 15:00	Session III
14:30 15:00	Applying Linked Open Data to Machine Translation for Cross-lingual Question Answering *Takaaki Matsumoto and Teruko Mitamura* Chronological and Geographical Measures for Evaluation of World History Essay QA in University Entrance Exams *Hideyuki Shibuki, Kotaro Sakamoto, Madoka Ishioroshi, Akira Fujita, Yoshinobu Kano, Teruko Mitamura, Tatsunori Mori and Noriko Kando*
15:00 - 15:30	Session IV (Demo session)
	Video Question Answering to Find a Desired Video Segment *Mayu Otani, Yuta Nakashima, Esa Rahtu and Janne Heikkilä* OKBQA Framework for collaboration on developing natural language question answering systems *Jin-Dong Kim, Christina Unger, Axel-Cyrille Ngonga Ngomo, Andre Freitas, Young-Gyun Hahm, Jiseong Kim, Sangha Nam, Gyu-Hyun Choi, Jeong-Uk Kim, Ricardo Usbeck, Myoung-Gu Kang and Key-Sun Choi* FelisCatusZero: A world history essay question answering for the University of Tokyo's entrance exam *Kotaro Sakamoto, Takaaki Matsumoto, Madoka Ishioroshi, Hideyuki Shibuki, Tatsunori Mori, Noriko Kando and Teruko Mitamura*
15:30 - 15:50	Coffee break
15:50 - 17:20	Joint panel discussion with KG4IR (in the KG4IR workshop room)
	Keynote speech *Hannah Bast, Universitaet Freiburg, Germany* Panel discussion *Hannah Bast, Universitaet Freiburg, Germany* *Noriko Kando, National Institute of Informatics, Japan* *Jaap Kamps, University of Amsterdam, The Netherlands* *Edgar Meij, Bloomberg L.P., U.K.* *Bogdan Arsintescu, LinkedIn, U.S.A* *David Carmel, Yahoo!, Israel*
17:20 - 17:30	Closing

Contents

Papers

Demos

Controlling Expressiveness of Question Interpretation with a Constrained Tree Transducer Induction

Pascual Martínez-Gómez
pascual.mg@aist.go.jp
Artificial Intelligence Research Center, AIST
Tokyo, Japan

Yusuke Miyao[*][†]
yusuke@nii.ac.jp
National Institute of Informatics
Tokyo, Japan

ABSTRACT

An important problem in Question Answering over Knowledge Bases is to *interpret a question* into a database query. This problem can be formulated as an instance of *semantic parsing* where a natural language utterance is analyzed into a (possibly executable) meaning representation. Most semantic parsing strategies for Question Answering use models with limited expressiveness because it is difficult to characterize it and systematically control it. In this work we use tree-to-tree transducers which are very general and solid models to transform the syntactic tree of a question into the executable semantic tree of a database query. When designing these tree transducers, we identify two parameters that influence the construction cost and their expressive capabilities, namely the tree fragment depth and number of variables of the rules. We characterize the search space of tree transducer construction in terms of these parameters and show considerable improvements in accuracy as we increase the expressive power.

KEYWORDS

Question Answering, Tree Transducers, Question Interpretation

1 INTRODUCTION AND RELATED WORK

Question Answering (QA) over Knowledge Bases (KBs) is a step forward in the realization of human-machine natural language interfaces to large structured knowledge resources. In this task, one of the main challenges is the interpretation of a natural language utterance into an executable meaning representation. In the community of Natural Language Processing, this task can be formulated as a semantic parsing problem where the objective is to produce a symbolic meaning representation with predicates grounded to Knowledge Base constants (entities and relations). This problem is different from that of the *Simple Questions* tasks [2, 21] where researchers try to identify a single fact from a KB given a short question that typically involves only one relation. Instead, we aim to answer questions that require higher levels of compositionality, aggregation or KB inference.

There are two main strategies when doing executable semantic parsing for QA over large KBs. The first one is that of Berant et al. [1] where a question is directly parsed into a semantic formula without an intermediate syntactic representation (string-to-tree transformations). In this approach, the grammar of the executable semantic representation is manually specified[1] and the parameters of a statistical model are estimated with the objective to guide the parser towards correct derivations. The second strategy follows the principles of the syntax-semantic interface, which is a popular paradigm of semantic compositionality among linguists and formal semanticists (tree-to-tree transformations). In this strategy, the syntactic analysis of a question (or more generally, a sentence) is used to guide the semantic composition of a symbolic meaning representation. Some early representatives are the work of Ge and Mooney [5] and that of Wong and Mooney [19]. Most recently, dependency trees of questions are transformed into grounded meaning representations (SPARQL queries) using a set of manually designed rules [14], establishing a new state of the art.

We commit to this second strategy with tree-to-tree transducers which are general and well-studied models [15, 17] that describe how input trees can be transformed into output trees. Knight and Graehl [10] give a good overview. Given the generality of these models, they have been used in a variety of text transformation tasks such as paraphrasing and textual entailment [20], text summarization [4] or question answering [8]. However, it is difficult to induce these tree transducers in semantic parsing tasks where there is a large vocabulary in the target language (i.e. number of constants in the KB) and typically small numbers of examples of tree pairs in the training set. Martínez-Gómez and Miyao [13] proposed a tree mapping algorithm that served as the basis to induce tree transducers from small data, allowing the application of these models to Question Answering tasks over large Knowledge Bases. However, they did not study how transducer rules with different expressiveness affect model accuracy in a downstream application and its impact in the transducer construction cost.

Our contribution is a formal characterization of the search space in the tree mapping algorithm that induces tree transducer grammars. This characterization evidences two critical parameters that control the model expressiveness and its complexity, which we believe is useful in text transformation tasks that deploy synchronous tree grammars. We evaluate the expressiveness of the resulting tree transducers in terms of QA accuracy and we demonstrate the importance of tree-to-tree transformation models whose rules consume and produce tree fragments of depth larger than one.

2 BACKGROUND

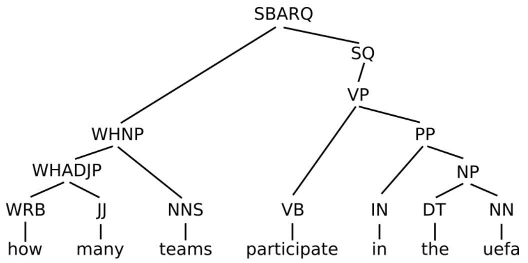
Given a question and a Knowledge Base, our system performs the following steps. First, obtain the constituent syntactic tree of the question. Second, use a set of weighted rules to transform fragments of the syntactic tree into fragments of a semantic tree and compose the executable meaning representation. Finally, execute

[*] Also with Artificial Intelligence Research Center, AIST, Tokyo, Japan.
[†] Also with The Graduate University for Advanced Studies (SOKENDAI), Tokyo, Japan.
[1] These grammars are typically small. Thus, manual specification is often feasible.

the meaning representation (i.e. SPARQL query) on a KB and return the results. As a running example, consider the question:

Q: *how many teams participate in the uefa*

which is syntactically analyzed into the following constituent tree:



(1)

The objective is to transform such a syntactic tree into the following SPARQL query:

$$\begin{array}{llll} \multicolumn{4}{l}{\texttt{SELECT COUNT(}?x\texttt{) WHERE \{}} \\ ?a & \texttt{Team} & ?x & . \\ ?a & \texttt{League} & \texttt{Uefa} & .\} \end{array} \tag{2}$$

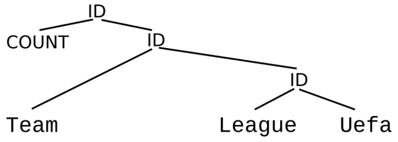
which corresponds to the following λ expression:

$$\texttt{count}(\lambda x.\exists a.\texttt{Team}(x,a) \wedge \texttt{League}(a,\texttt{Uefa})) \tag{3}$$

where Team and Uefa are the KB entities to which the natural language expressions *teams* and *uefa* map into. Note that the expressions in Equation 2 and Equation 3 do not have a tree structure but a graph structure due to the presence of repeated variables ($?a$ or a) at the leaves. However, it is convenient to shape these graph structures into the form of a tree. To this end, Liang [11] proposes the λ-DCS tree language, where existentially quantified variables are made implicit. For our running example, the λ-DCS expression would be:

$$\texttt{count(Team.League.Uefa)} \tag{4}$$

which can be trivially represented as a semantic tree structure:



(5)

Thus, we transform the syntactic tree s in (1)[2] into the executable semantic tree t in (5)[3] which can be later trivially converted into the SPARQL query in (2).

The tree-to-tree transformation from (1) to (5) can be performed with a set of weighted rules (see Figure 1) whose left-hand-sides match and consume fragments of the syntactic tree (1) and produce tree fragments of the executable semantic tree (5). These rules are at the core of a tree transducer, which we describe now.

Following the same terminology as Graehl and Knight [7], a tree transducer is a 5-tuple $(Q, \Sigma, \Delta, q_{\text{start}}, \mathcal{R})$ where Q is the set of transducer states that carry some memory through the transformation process, Σ is the set of input symbols (i.e. syntactic categories and English words), Δ is the set of output symbols (KB entities and relations), q_{start} is the initial state from which the tree transformation starts, and $\mathcal{R}$ is the set of transducer rules. Transducer rules $r_i \in \mathcal{R}$ define atomic transformations and they have the form $q.t_i \overset{s}{\rightarrow} t_o$, where $q \in Q$ is the rule state, t_i is an input (syntactic) tree fragment, t_o is an output (semantic) tree fragment, and s is the score of the rule. In our work, we commit to *extended*[4] *root-to-frontier*[5] *linear*[6] transducers [12], possibly with *deleting*[7] operations. Some rules are *terminal rules* whose t_i match entire syntactic subtrees and whose t_o produce semantic subtrees (e.g. $r4$ and $r5$). Other rules (e.g. $r1 - r3$) are *non-terminal* rules where variables x_i are connection points with other rules thus carrying over the tree compositionality.

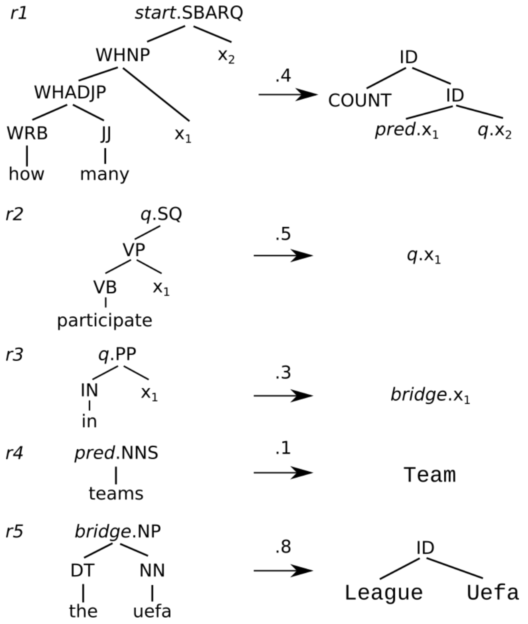


Figure 1: Transducer rules that transform syntactic tree (1) into executable semantic tree (5).

Note in Figure 1 how some rules are more complex (and expressive) than others. For instance, the left-hand-side of $r4$ is a syntactic subtree of depth 1 (one-level subtree) with no variables, whereas the left-hand-side of $r1$ has depth 4 and two variables. We formalize this concept in the next section and characterize the space of possible rules by parameterizing it in terms of rule depth and number of variables.

3 METHODOLOGY

In our characterization, the expressiveness of a tree transducer depends on the expressiveness of its rules since we keep the states Q and input/output vocabulary (Σ and Δ) constant. In turn, the expressiveness of the transducer rules ($r = q.t_i \overset{s}{\rightarrow} t_o$) depends on the characteristics of the input and output tree fragments (t_i and t_o). Thus, if we want to characterize and parameterize the space of

[2] We also call it *source* tree s.

[3] We also call it *target* tree t.

[4] t_i may have depth larger than 1.

[5] Top-down transformations.

[6] t_i variables appear at most once in the t_o.

[7] Some variables on the t_i may not appear in the t_o.

induced transducers we need to characterize the space of possible tree fragments. To this end, we need to introduce terminology that allows us to define tree fragments with precision.

We uniquely identify nodes in a tree by using paths $p \in \mathcal{P}$, which are similar to Gorn addresses [6] but with a tuple notation. For example, in Figure 1, the path $p = (0)$ identifies the node with syntactic category WHNP, that is, the child index 0; the path $p = (0, 1)$ identifies the node NNS and $p = ()$ identifies the root. For convenience we define the path concatenation operation as $p_1 \cdot p_2$. For example, given $p_1 = (a, b)$ and $p_2 = (c, d)$, their concatenation results in $p_1 \cdot p_2 = (a, b, c, d)$ with a path length $|(a, b, c, d)| = 4$.

We define a tree fragment $t \in \mathcal{T}$ as $s \downarrow p \perp \{q_1, \ldots, q_n\}$, which is a tree fragment from tree s rooted at path $p \in \mathcal{P}$ with n variables substituting subtrees at subpaths $q_i \in \mathcal{P}$ for $1 \leq i \leq n$. Note that different orders of $\{q_1, \ldots, q_n\}$ allow to describe tree transformations that swap branches. The path p is a prefix of all q_i and q_i is not the prefix of any other subpath q_j for $i \neq j$ since variables can only appear at the leaves of tree fragments (no variable can have children). In our running example, the right-hand-side of $r1$ would be a tree fragment $t \downarrow () \perp \{(1, 0), (1, 1)\}$ where t is the target (semantic) tree, $p = ()$ since the rule starts at the root of t and there are two subpaths $q_1 = (1, 0)$ and $q_2 = (1, 1)$ that specify the location of each of the two variables x_1 and x_2. Another example would be the left-hand-side of $r4$, described as $s \downarrow (0, 1) \perp \{\}$ (note that there are no variables).

We can now define the space of tree fragments of a tree s rooted at any node p_i as:

$$\begin{aligned}\mathcal{T}^s_{p_i} = \{s \downarrow p_i \perp \{q_1, \ldots, q_n\} \mid \\ q_i \in \mathcal{P}_s \wedge p_i \cdot r = q_i \wedge |r| \leq d, \\ 1 \leq i \leq n\}\end{aligned} \tag{6}$$

where $\mathcal{P}_s$ is the set of paths to all nodes in tree s. The space of tree fragments $\mathcal{T}^s_{p_i}$ is parameterized by i) the maximum depth d of tree fragments ($|r| \leq d$) which controls the exponential growth and ii) the maximum number of variables n which limits the factorial combinations ($n!$) of branch orderings.

The space of transducer rules[8] is formed by all pairs of tree fragments $\mathcal{T}^s_{p_i} \times \mathcal{T}^t_{p_o}$ for all $p_i \in \mathcal{P}_i$ paths in the syntactic source tree s and all $p_o \in \mathcal{P}_o$ paths in the semantic target tree t. The mapping cost between between trees s and t at paths p_i and p_o can be computed as:

$$\begin{aligned}&\mathrm{C}\left(s \downarrow p_i, t \downarrow p_o\right) = \\ &\min_{\mathbf{q},\mathbf{q}'}\{\gamma\left(s \downarrow p_i \perp \mathbf{q}, t \downarrow p_o \perp \mathbf{q}'\right) + \sum_{j=1}^{|\mathbf{q}|} \mathrm{C}\left(s \downarrow q_j, t \downarrow q'_j\right)\}\end{aligned} \tag{7}$$

where $s \downarrow p_i \perp \mathbf{q} \in \mathcal{T}^s_{p_i}$, $t \downarrow p_o \perp \mathbf{q}' \in \mathcal{T}^t_{p_o}$, $q_j \in \mathbf{q}$ and $q'_j \in \mathbf{q}'$. The cost between two tree fragments $\gamma(t_i, t_o)$ depends on the application. In our case, it is an ensemble of cost functions that assigns low costs to pairs of syntactic and semantic subtrees whose leaves (natural language phrases and KB constants) may have a linking relation.

The mapping cost between the roots of s and t can be computed as $\mathrm{C}(s \downarrow (), t \downarrow ())$ whereas the node-to-node correspondences can be recovered using back-pointers as it is usual in dynamic programming. For the sake of efficiency we perform the search using an approximate bottom-up beam-search algorithm [13] parameterized by d and n.

4 EXPERIMENTS

We use tree transducers to transform the syntactic tree of a question into a SPARQL query. We evaluate on Free917, a corpus of 641 question-query pairs for training and 276 questions for testing. We obtain syntactic constituent trees of questions using the Stanford caseless models [9] which produce trees with an average of 24.5 nodes and tree height 7.4 in this dataset. The gold queries in this dataset typically have between one and three statements, possibly with a *count* aggregator. We use the entity lexicon released by Cai and Yates [3] and the relation lexicon released by Martínez-Gómez and Miyao [13]. Our KB is the same Freebase dump as in Berant et al. [1], which contains millions of facts.

We automatically induce (construct) tree-to-tree transducers by extracting rules from pairs of question syntactic trees and query trees. This rule extraction is performed by a tree mapping search algorithm that is constrained by d and n, thus resulting in tree transducers with different levels of expressiveness[9]. The weights (parameters) of the resulting tree-to-tree transducers are estimated using the latent variable averaged structured perceptron. In this parameter estimation routine, rules are represented by a feature vector[10] and the rule score is the result of a linear combination between these rule features and model weights. We reward[11] weights associated to rule sequences that transform a question syntactic tree into a query that retrieves the correct answer as given in the gold training data. We perform 3 iterations over the whole training set and use the learned weights for the decoding stage. The number of iterations and the learning rate are estimated on a validation split from the training data.

For questions in the test set, our decoder generates 10,000 target trees which we trivially convert into SPARQL queries that we run against the KB. Then, we keep those that retrieve at least one answer. We count a point of accuracy if the highest scoring SPARQL query retrieves the correct set of answers whereas we count a point of coverage if at least one query in the 10,000 candidates retrieves the correct set of answers. Then we average the accuracy and coverage over the whole test set.

Our results in terms of accuracy and coverage for different settings of d and n are in Table 1. The table also displays the average number of rules extracted per tree pair in the training data and the average time, median, maximum and standard deviation of the tree mapping and rule extraction across all training tree pairs. The system `t2t-d∞-n∞` imposes no constraints on d and n whereas the rest of the systems do. For example, the system `t2t-d3-n∞` limits the tree fragment depth to $d \leq 3$ but imposes no constraints on the number of variables ($n \leq \infty$).

[8] If we ignore the rule states.

[9] That is, transducers with rules with a maximum of d tree fragment depth and a maximum of n variables.

[10] These features are instantiated using hand-engineered feature templates that measure rule characteristics such as the number of nodes in the left- or right-hand-side tree fragments, the presence of an aggregator function, n-gram overlap between words in the question and text literals associated to KB entities or relations, etc.

[11] That is, we increase their value by a small factor which is the learning rate: 0.01.

Systems	Acc.	Cov.	# Rules	Time
t2t-d∞-n∞	.64	.79	708	3.9, 1.7, 166.6, 8.6
t2t-d1-n∞	.36	.66	1958	1.3, 0.9, 16.5, 1.1
t2t-d2-n∞	.56	.84	886	1.7, 1.1, 24.3, 2.0
t2t-d3-n∞	.65	.80	746	2.4, 1.3, 45.7, 3.5
t2t-d4-n∞	.64	.79	713	2.8, 1.4, 67.0, 4.7
t2t-d5-n∞	.64	.79	708	3.0, 1.4, 87.1, 5.5
t2t-d∞-n1	.09	.30	1228	3.0, 2.0, 44.0, 3.3
t2t-d∞-n2	.63	.83	743	3.4, 1.9, 88.5, 5.3
t2t-d∞-n3	.63	.79	713	3.6, 1.8, 128.5, 6.9
t2t-d∞-n4	.63	.78	706	4.2, 1.9, 149.8, 8.6
t2t-d∞-n5	.63	.78	708	4.1, 1.9, 154.0, 8.3

Table 1: Accuracy and coverage results for the test split of Free917. "# Rules" and "Time" stand for the average number of rules and tree mapping time (average, median, maximum and standard deviation) across tree pairs in the training set.

When inducing transducers with simple rules (depth $d \leq 2$ or $n = 1$), the accuracy and coverage is low but the tree mapping is fast. The accuracy (and coverage) increases progressively and saturates between .63 and .65 (.78 and .84) as we increase the rule expressiveness by setting higher limits in tree fragment depth and number of variables. However, tree mapping time also seems to increase proportionally to rule expressiveness in terms of d and n but no asymptotic trend can be observed due to the small training set and relatively small question complexity. The average number of rules changes only slightly for $d \geq 4$ and $n \geq 3$, which suggests that questions in Free917 do not require transformation operations more expressive than that.

As a comparison to other systems, SEMPRE [1] obtains and accuracy of .62, whereas Reddy et al. [14]'s DepLambda system obtains an accuracy of .78 and a coverage of .96. However, these systems are not directly comparable because they use different entity/relation linkers, manually specified grammars and (or) hand-crafted rules.

5 FUTURE WORK AND CONCLUSION

We have concentrated on describing the search space of the question interpretation problem but we have neglected the grounding problem (mapping natural language expressions to KB constants) by re-using manually created lexicons of entities and relations. However, in the realization of a full-fledged QA system we need to integrate wide-coverage entity and relation linkers which we plan to do in the near future. Moreover, we claimed that tree transducers are general models but we evaluated on a single dataset. Our next step is to use other datasets such as WebQuestions [1], GraphQuestions [16] and QALD challenges [18] to assess the generality of these models. Yet another extension is to use these tree-to-tree transducers as an effective generalization of symbolic semantic parsing with grounding. This extension would encompass the current state of the art based on manually designed transformation rules for dependency trees but with the added advantage of including a bootstrapping mechanism to acquire new rules for questions with previously unseen syntactic structures.

We showed a characterization of the tree mapping search space that is parameterized by the tree fragment depth d and number of variables n in rules. Experimental results showed how these two parameters trade QA accuracy by tree mapping time. Specifically, we found that tree transducers whose rules are limited to $d \leq 2$ or $n = 1$ obtain a low accuracy but the tree mapping time to induce these transducers is fast. Higher accuracies were obtained when using more expressive rules ($d = 3$ or $n = 2$). However, no further gains were obtained for larger tree fragment depths and number of variables because the questions and target trees in the Free917 corpus are relatively simple and do not require induced rules with such a high level of expressiveness. As a comparison, SEMPRE and DepLambda have grammar rules of depth $d = 1$. The results in this paper suggest that those systems could also be improved by increasing their rule depth. However, since their grammars are hand-crafted, the manual specification of these complex rules is not trivial and requires very fine grained linguistic analysis.

ACKNOWLEDGMENTS

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO), and is also supported by JSPS KAKENHI Grant Number 16K16111. We thank Yoshimasa Tsuruoka and the anonymous reviewers for their helpful comments.

REFERENCES

[1] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, 1533–1544. http://www.aclweb.org/anthology/D13-1160

[2] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale Simple Question Answering with Memory Networks. *CoRR* abs/1506.02075 (2015). http://arxiv.org/abs/1506.02075

[3] Qingqing Cai and Alexander Yates. 2013. Large-scale Semantic Parsing via Schema Matching and Lexicon Extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, 423–433. http://www.aclweb.org/anthology/P13-1042

[4] Trevor Anthony Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research* 34 (2009), 637–674.

[5] Ruifang Ge and Raymond J. Mooney. 2005. A Statistical Semantic Parser That Integrates Syntax and Semantics. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CONLL '05)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 9–16. http://dl.acm.org/citation.cfm?id=1706543.1706546

[6] Saul Gorn. 1965. Explicit definitions and linguistic dominoes. In *Systems and Computer Science, Proceedings of the Conference held at Univ. of Western Ontario*. 77–115.

[7] Jonathan Graehl and Kevin Knight. 2004. Training Tree Transducers. In *HLT-NAACL 2004: Main Proceedings*, Daniel Marcu Susan Dumais and Salim Roukos (Eds.). Association for Computational Linguistics, Boston, Massachusetts, USA, 105–112.

[8] Bevan Keeley Jones, Mark Johnson, and Sharon Goldwater. 2012. Semantic Parsing with Bayesian Tree Transducers. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1 (ACL '12)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 488–496. http://dl.acm.org/citation.cfm?id=2390524.2390593

[9] Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sapporo, Japan, 423–430. https://doi.org/10.3115/1075096.1075150

[10] Kevin Knight and Jonathan Graehl. 2005. An Overview of Probabilistic Tree Transducers for Natural Language Processing. In *Computational Linguistics and*

Intelligent Text Processing, Alexander Gelbukh (Ed.). Lecture Notes in Computer Science, Vol. 3406. Springer Berlin Heidelberg, 1–24. https://doi.org/10.1007/978-3-540-30586-6_1

[11] Percy Liang. 2013. Lambda Dependency-Based Compositional Semantics. *CoRR* abs/1309.4408 (2013). http://arxiv.org/abs/1309.4408

[12] Andreas Maletti, Jonathan Graehl, Mark Hopkins, and Kevin Knight. 2009. The Power of Extended Top-Down Tree Transducers. *SIAM J. Comput.* 39, 2 (2009), 410–430. https://doi.org/10.1137/070699160

[13] Pascual Martínez-Gómez and Yusuke Miyao. 2016. Rule Extraction for Tree-to-Tree Transducers by Cost Minimization. In *Proc. of EMNLP*. 12–22.

[14] Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming Dependency Structures to Logical Forms for Semantic Parsing. *Transactions of the ACL* 4 (2016), 127–140.

[15] William C. Rounds. 1970. Mappings and grammars on trees. *Mathematical systems theory* 4, 3 (1970), 257–287. https://doi.org/10.1007/BF01695769

[16] Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gur, Zenghui Yan, and Xifeng Yan. 2016. On Generating Characteristic-rich Question Sets for QA Evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 562–572. https://aclweb.org/anthology/D16-1054

[17] James W. Thatcher. 1970. Generalized sequential machine maps. *J. Comput. System Sci.* 4, 4 (1970), 339 – 367. https://doi.org/10.1016/S0022-0000(70)80017-4

[18] Christina Unger, Corina Forascu, Vanessa Lopez, Axel-Cyrille Ngonga Ngomo, Elena Cabrio, Philipp Cimiano, and Sebastian Walter. 2015. Question Answering over Linked Data (QALD-5). In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum*, Linda Cappellato, Nicola Ferro, Gareth Jones, and Eric San Juan (Eds.), Vol. 1391. Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum.

[19] Yuk Wah Wong and Raymond J. Mooney. 2006. Learning for Semantic Parsing with Statistical Machine Translation. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL '06)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 439–446. https://doi.org/10.3115/1220835.1220891

[20] Dekai Wu. 2005. Recognizing Paraphrases and Textual Entailment Using Inversion Transduction Grammars. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment (EMSEE '05)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 25–30. http://dl.acm.org/citation.cfm?id=1631862.1631867

[21] Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. 2016. Simple Question Answering by Attentive Convolutional Neural Network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, 1746–1756. http://aclweb.org/anthology/C16-1164

TaxoPhrase: Exploring Knowledge Base via Joint Learning of Taxonomy and Topical Phrases

Weijing Huang
Key Laboratory of High Confidence Software Technologies
(Ministry of Education), EECS, Peking University
huangwaleking@gmail.com

Wei Chen*
Key Laboratory of High Confidence Software Technologies
(Ministry of Education), EECS, Peking University
pekingchenwei@pku.edu.cn

Tengjiao Wang
Key Laboratory of High Confidence Software Technologies
(Ministry of Education), EECS, Peking University
tjwang@pku.edu.cn

Shibo Tao
SPCCTA, School of Electronics and Computer
Engineering, Peking University
expo.tao@gmail.com

ABSTRACT

Knowledge bases restore many facts about the world. But due to the big size of knowledge bases, it is not easy to take a quick overview onto their restored knowledge. In favor of the taxonomy structure and the phrases in the content of entities, this paper proposes an exploratory tool TaxoPhrase on the knowledge base. TaxoPhrase (1) is a novel Markov Random Field based topic model to learn the taxonomy structure and topical phrases jointly; (2) extracts the topics over subcategories, entities, and phrases, and represents the extracted topics as the overview information for a given category in the knowledge base. The experiments on the example categories *Mathematics*, *Chemistry*, and *Argentina* in the English Wikipedia demonstrate that our proposed TaxoPhrase provides an effective tool to explore the knowledge base.

KEYWORDS

Knowledge Base, Exploratory Tool, Topical Phrases, Taxonomy Structure, Topic Model, Markov Random Field

1 INTRODUCTION

Knowledge bases[5][3][10][13] are constructed elaborately to restore the information representing the facts about the world. And due to the big size of knowledge bases, it's necessary to provide an exploratory tool to take a quick overview on them. For example, there are more than 5.3 million articles in Wikipedia (March 2017)[1], far beyond the scale that human can read all. A suitable exploratory tool benefits the users of the knowledge base to have an overall perspective of the restored knowledge.

We attempt to achieve this purpose by answering the following three questions: (1) Q1, what are the main subtopics related to a given topic in knowledge base; (2) Q2, what are the related entities for each subtopic; (3) Q3, what are the key summarization corresponding to these subtopics. As many knowledge bases are constructed based on the Wikipedia, such as YAGO[10] and DBPeida[13], we answer the above three questions on Wikipedia without loss of generality. For this reason, in this paper the terms *page* and *entity* are used interchangeably.

Since categories in the knowledge base are used to group entities into similar subjects and are further organized hierarchically into the taxonomy, it seems straightforward to utilize the taxonomy to answer Q1 and Q2. But the size of taxonomy is too large to provide a quick overview for the whole knowledge base. Taking the English Wikipedia's taxonomy as an example, there are about 1.5 million category nodes. And the distribution of these categories' degrees is unbalanced, as shown in Figure 1. It means some categories contain too many subcategories, such as *Albums_by_artist*, whose out degree is 17,749; while most categories have very few subcategories, such as *Politics_of_Morelos* containing only one subcategory *Morelos_elections*. These characteristics indicate that the taxonomy structure is a large scale-free network[1]. So it's not easy to answer Q1 and Q2 directly only by the taxonomy structure.

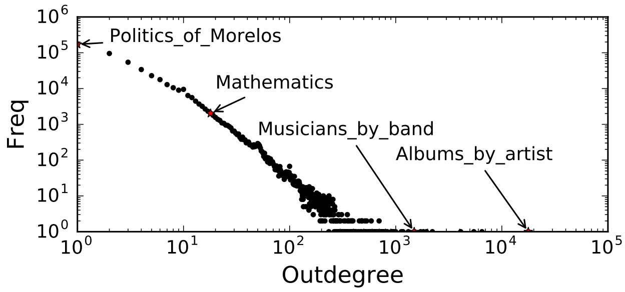


Figure 1: Distribution of categories' out degrees to subcategories, according to the study on the English Wikipedia.

Besides, the topic model[2], especially its extension on phrases, such as [6][9], is developed for the exploratory analysis on text corpora, and suitable for answering Q3. Usually the most frequent words or phrases in topics are used for summarizing the corpus[8]. However the meanings of the learned topics need to be manually interpreted[4], which may limit the usability of existing methods on Q3.

In this paper, we propose a novel exploratory tool TaxoPhrase, which learns the taxonomy structure and topical phrases in knowledge base jointly, and makes the questions Q1, Q2, Q3 tractable in a unified framework. The joint learning algorithm of TaxoPhrase is inspired by the complementary relation among the three parts in knowledge base: the categories in the taxonomy, the entities, and the phrases in entities' contents.

*Corresponding author.

[1] https://stats.wikimedia.org/EN/TablesWikipediaEN.htm

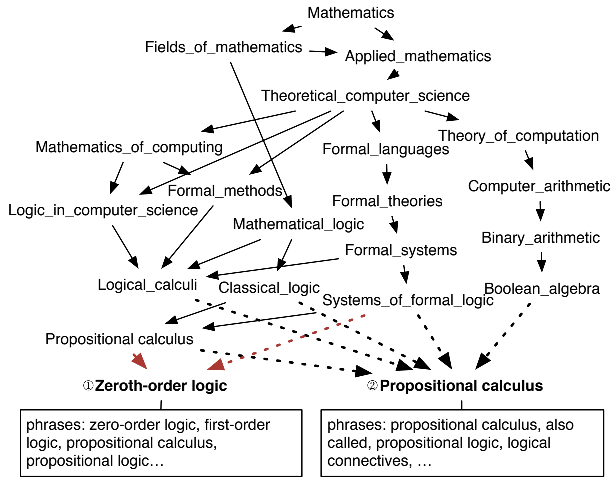


Figure 2: The illustration on the complementary relation among the three parts of knowledge base: the categories in the taxonomy, the entities, and the phrases in contents.

We take two examples to illustrate this kind of complementary relation, as shown in Figure 2. For the example ①, the Wikipedia page `Zeroth-order logic` directly belongs to two categories *Propositional_calculus* and *Systems_of_formal_logic*, which are descendant subcategories of the category *Mathematics*. The content of this page contains the phrases such as "zeroth-order logic", "first-order logic", "propositional calculus", and etc.. For the example ②, the Wikipedia page `Propositional calculus` belongs to five categories *Logical_calculi*, *Classical_logic*, *Propositional_calculus*, *Systems_of_formal_logic*, and *Boolean_algebra*, and contains the phrases "propositional calculus", "propositional logic", "logical connectives", and etc.. Obviously, these two pages share the similar categories in the taxonomy and the phrases in the content. Therefore, these two pages are more likely to correspond to the same subtopic *Mathematical_logic* under the category *Mathematics*. This fact is beneficial to answer Q1 and Q2. Meanwhile, the phrases shared by these two pages, e.g. "propositional calculus" and "propositional logic", are more likely to be grouped together as the topical phrases for the subtopic *Mathematical_logic*. These topical phrases are further used to give the answer of Q3.

To utilize the complementary relation among the three parts in the knowledge base, we extract the phrases and the related categories for each entity, and model them together in our proposed topic model TaxoPhrase.

To sum up, the contribution of our proposed TaxoPhrase is mainly in two aspects. (1) It is a novel Markov Random Field based topic model to learn the taxonomy structure and topical phrases jointly. (2) It extracts the topics over subcategories, entities, and phrases, and the extracted topics function as the overview information for a given category in the knowledge base. Furthermore, the experiments on example categories *Mathematics*, *Chemistry*, and *Argentina* in English Wikipedia demonstrate that our proposed method TaxoPhrase provides an effective tool to explore the knowledge base.

2 RELATED WORKS

To the best of our knowledge, there's few work on providing an explorative tool for the knowledge base. The most closest work to our motivation is Holloway's analyzing and visualizing the semantic coverage of Wikipedia[16], which visualizes the category network in two dimensions by the layout algorithm DrL (used to be VxOrd)[14]. However the layout doesn't provide the overview information of the knowledge base directly. The other works related to our approach can be grouped into two groups, which are taxonomy related and topical phrases related.

The taxonomy related works mainly focus on how to use the taxonomy of the knowledge base to enhance the quality of text mining tasks, such as Twixonomy[7], LGSA[11], and TransDetector[12].

The phrase related works extend the topic model to phrase level, such as ToPMine[6] and TPM[9]. ToPMine firstly extracts phrases by the frequent pattern mining on corpus, and secondly mine topical phrases on the "bag of phrases", in which the single word is treated as the shortest phrase. TPM reuses the phrases generated from ToPMine. We follow it and use the phrases as the input of our tool. Considering the big size of the knowledge base, we run the LDA[2] on phrases as a baseline, rather than running the full ToPMine.

3 PROPOSED METHOD

3.1 Preprocessing

There are two parts to be extracted for each entity. The first are phrases, that are generated from ToPMine[6]. The second are the category-information, a.k.a., the set of *category* → *subcategory* edges related to the entity, defined formally in Definition 1.

Definition 1 (Category-Information). Given the entity d, the category information is a set of edges $s_{dn'} \rightarrow t_{dn'}$, where $t_{dn'}$ is the direct subcategory of $s_{dn'}$ and they are both the ancestors of the entity d. And we denote d's category-information as $\mathbf{c}_d = \{(s_{dn'}, t_{dn'})\}_{n'=1}^{N'_d}$.

For instance, all the categories in Figure 2 are the ancestors of the entity `Propositional calculus`, so all the 25 edges in Figure 2 are included in its category-information. As suggested by [12], we prune the cycles according to nodes' PageRank score to make the taxonomy as a Directed Acyclic Graph. We extract the category-information for a given entity on the taxonomy DAG.

3.2 TaxoPhrase

In this subsection, we present the model TaxoPhrase, which is illustrated in Figure 3. Since the phrases and category-information for each entity are already generated in the preprocessing phase, we treat them as the input of the TaxoPhrase model.

Joint Learning. Same as the traditional probabilistic topic models such as LDA[2], we assume that there are K topics in TaxoPhrase, and for each entity d we use the K-dimension vector θ_d to represent its latent topic distribution. The difference is that we model the categories and the phrases jointly. We denote the topics as $\{(\phi_k^{(\tau)}, \phi_k)\}_{k=1}^{K}$, where $\phi_k^{(\tau)}$ is the category distribution on the k-th topic, and ϕ_k is the word distribution on the k-th topic.

We connect the generation process of the phrases and the category-information by the entity-topic distribution θ_d. For each entity d, the input data include the category-information $\mathbf{c}_d = \{(s_{dn'}, t_{dn'})\}_{n'=1}^{N'_d}$

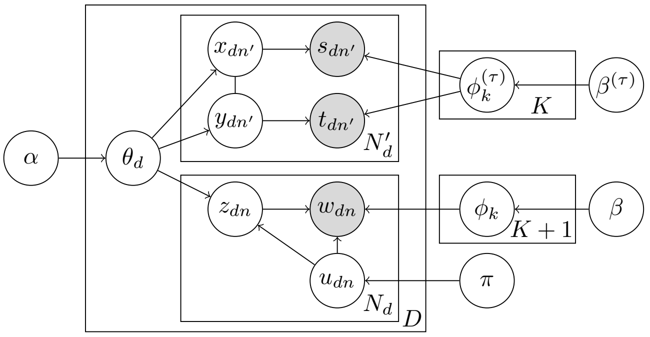


Figure 3: Illustration of the model TaxoPhrase

and the phrases $\boldsymbol{w_d} = \{w_{dn}\}_{n=1}^{N_d}$. We use the discrete values $\boldsymbol{x_d} = \{x_{dn'}\}_{n'=1}^{N'_d}$, $\boldsymbol{y_d} = \{y_{dn'}\}_{n'=1}^{N'_d}$ to represent the hidden topics of the category nodes $\boldsymbol{s_d}$ and $\boldsymbol{t_d}$ respectively. Correspondingly, we use the discrete values $\boldsymbol{z_d}$ for the phrases $\boldsymbol{w_d}$. The discrete topic assignments $\boldsymbol{x_d}$, $\boldsymbol{y_d}$, and $\boldsymbol{z_d}$ are all drawn from the same distribution $Multinomial(\theta_d)$, and are impacted by each other. Thus, the topic-category distribution $\phi_k^{(\tau)}$ and the topic-phrase distribution ϕ_k are aligned with each other.

Additionally, we use the background topic ϕ_0 to model the high frequent background phrases to enhance the topic modeling quality. The switcher variable u_{dn} is introduced for determining whether the phrase w_{dn} belongs to the background topic.

Markov Random Field on the taxonomy. As mentioned in Section 1, categories connected via an edge in the taxonomy tend to share the similar topic. Given the category edges $(s_{dn'}, t_{dn'})$, we put their latent topic assignments $x_{dn'}$ and $y_{dn'}$ into a Markov Random Field to capture this tendency. Specifically, we define the binary potential $\exp(\mathbb{I}(x_{dn'} = y_{dn'}))$ to encourage $x_{dn'}$ to have the same topic as $y_{dn'}$, where $\mathbb{I}(.)$ is the indicator function. And we use the unary potential $p(x_{dn'}|\theta_d)$ to link θ_d and $x_{dn'}$, which is defined by the multinomial distribution with the parameter θ_d as $p(x_{dn'}|\theta_d) = \prod_{k=1}^{K} \theta_{dk}^{\mathbb{I}(x_{dn'}=k)}$. The unary potential $p(y_{dn'}|\theta_d)$ is defined in the same way to link the entity's topic distribution θ_d and the topic assignment $y_{dn'}$ of the category $t_{dn'}$.

Because of the joint learning for categories and phrases, the entity's topic distribution θ_d also links with the phrases' topic assignments $\boldsymbol{z_d}$, which are generated with the probability $p(z_{dn}|\theta_d, u_{dn} = 1) = \prod_{k=1}^{K} \theta_{dk}^{\mathbb{I}(z_{dn}=k)\mathbb{I}(u_{dn}=1)}$. Therefore, the topic assignments share the following joint distribution.

$$p(\boldsymbol{x_d}, \boldsymbol{y_d}, \boldsymbol{z_d}|\theta_d, \boldsymbol{u_d}) = \frac{1}{A_d(\theta_d)} \prod_{n=1}^{N_d} p(z_{dn}|\theta_d, u_{dn}) \prod_{n'=1}^{N'_d} p(x_{dn'}|\theta_d) \cdot \prod_{n'=1}^{N'_d} p(y_{dn'}|\theta_d) \exp\{\sum_{n'=1}^{N'_d} \mathbb{I}(x_{dn'} = y_{dn'})\} \tag{1}$$

In Equation (1), A_d is the partition function to normalize the joint distribution. According to the Equation (1), the topic assignments do not only depends on the entity's topic distribution θ_d, but also depends on the other topic assignments in the Markov Random Field.

Generation Process. To sum up, given the hyper parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\beta^{(\tau)}}$, π, and the number of topics K, the generation process of the entities in the knowledge base can be described as follows.

1. Draw phrases' background topic $\boldsymbol{\phi_0} \sim Dir(\boldsymbol{\beta})$.
2. For each topic $k \in \{1, \cdots, K\}$,
   (a) draw phrase distribution on the topic $\boldsymbol{\phi_k} \sim Dir(\boldsymbol{\beta})$,
   (b) draw category distribution on the topic $\boldsymbol{\phi_k^{(\tau)}} \sim Dir(\boldsymbol{\beta^{(\tau)}})$.
3. For each entity index $d \in \{1, \cdots, D\}$,
   (a) draw the topic distribution on the entity $\boldsymbol{\theta_d} \sim Dir(\boldsymbol{\alpha})$,
   (b) for each phrase index $n \in \{1, \cdots, N_d\}$,
      (i.) draw the switcher $u_{dn} \sim Bernoulli(\pi)$,
   (c) draw topic assignments $\boldsymbol{x_d}$, $\boldsymbol{y_d}$, and $\boldsymbol{z_d}$ according to the Equation (1),
   (d) for each phrase index $n \in \{1, \cdots, N_d\}$,
      (i.) if $u_{dn} = 0$ draw the phrase $w_{dn} \sim Multinomial(\boldsymbol{\phi_0})$, else draw the phrase $w_{dn} \sim Multinomial(\boldsymbol{\phi_{z_{dn}}})$,
   (e) for $\boldsymbol{c_d}$'s each category edge index $n' \in \{1, \cdots, N'_d\}$,
      (i.) draw the category $s_{dn'} \sim Multinomial(\boldsymbol{\phi^{(\tau)}_{x_{dn'}}})$,
      (ii.) draw the category $t_{dn'} \sim Multinomial(\boldsymbol{\phi^{(\tau)}_{y_{dn'}}})$.

Inference. Firstly, we joint sample for z_{dn} and u_{dn} together according to the Equation (2) and (3), as z_{dn} is meaningful only when the switcher variable u_{dn} is set to 1. The sampling result of z_{dn} and u_{dn} depends on three parts, the entity's topic distribution $n_{d,k}$, the phrase's topic distribution $n_{k,v}$ and $n_{B,v}$, and the coin toss π. The sampling result also takes the impact from the categories into consideration, because $n_{d,k} = \sum_{n=1}^{N_d} \mathbb{I}(z_{dn} = k)\mathbb{I}(u_{dn} = 1) + \sum_{n'=1}^{N'_d} \mathbb{I}(x_{dn'} = k) + \sum_{n'=1}^{N'_d} \mathbb{I}(y_{dn'} = k)$.

$$p(z_{dn} = k, u_{dn} = 1|\boldsymbol{z_{\neg dn}}, \boldsymbol{u_{\neg dn}}, \boldsymbol{x}, \boldsymbol{y}, w_{dn} = v, \boldsymbol{w_{\neg dn}}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\beta^{(\tau)}}, \pi) \propto \frac{n_{d,k} + \alpha_k}{\sum_{k=1}^{K} n_{d,k} + \sum_{k=1}^{K} \alpha_k} \cdot \frac{n_{k,v} + \beta_v}{\sum_{v=1}^{V} n_{k,v} + \sum_{v=1}^{V} \beta_v} \cdot \pi \tag{2}$$

$$p(u_{dn} = 0|\boldsymbol{z_{\neg dn}}, \boldsymbol{u_{\neg dn}}, \boldsymbol{x}, \boldsymbol{y}, w_{dn} = v, \boldsymbol{w_{\neg dn}}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\beta^{(\tau)}}, \pi) \propto \frac{n_{B,v} + \beta_v}{\sum_{v=1}^{V} n_{B,v} + \sum_{v=1}^{V} \beta_v} \cdot (1 - \pi) \tag{3}$$

Secondly, we sample for x_{dn} and y_{dn} sequentially as the Equation (4). The exponential part $\exp\{\mathbb{I}(y_{d,n} = k)\}$ encourages that x_{dn} is sampled with the same topic as y_{dn}. That's where the Markov Random Field plays the role on the taxonomy structure.

$$p(x_{dn} = k|\boldsymbol{x_{\neg dn}}, \boldsymbol{y}, s_{dn} = v^{(\tau)}, \boldsymbol{s_{\neg dn}}, \boldsymbol{t}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\beta^{(\tau)}}, \pi) \propto (n_{d,k} + \alpha_k) \frac{n^{(\tau)}_{k,v^{(\tau)}} + \beta^{(\tau)}_{v^{(\tau)}}}{\sum_{v^{(\tau)}=1}^{V^{(\tau)}} n^{(\tau)}_{k,v^{(\tau)}} + \sum_{v^{(\tau)}=1}^{V^{(\tau)}} \beta^{(\tau)}_{v^{(\tau)}}} \exp\{\mathbb{I}(y_{d,n} = k)\} \tag{4}$$

The sampling equation for y_{dn} is symmetric with x_{dn}'s as they are symmetric in the Markov Random Field.

Table 1: Top 5 topics learned by TaxoPhrase. The line in the *italic font* indicates the categories.

Topic 1	*Mathematics_awards, Mathematicians_by_award, Mathematicians_by_nationality, Mathematicians_by_field*
(Entities)	`John Cedric Griffiths Teaching Award`, `Santosh Vempala`, `Aisenstadt Prize`, `Subhash Suri`, `David P. Dobkin`
(Phrases)	university of california, american mathematical society, professor of mathematics, princeton university, computer science, harvard university, american mathematician, stanford university, massachusetts institute of technology, columbia university
Topic 2	*Geometry_stubs, Differential_geometry_stubs, Elementary_geometry_stubs, Polyhedron_stubs*
(Entities)	`Enneadecahedron`, `Icosahedral pyramid`, `Expanded icosidodecahedron`, `Pentadecahedron`, `Cubic cupola`
(Phrases)	three dimensional, platonic solids, johnson solids, uniform polyhedron compound, symmetry group, regular dodecahedron, triangular faces, vertex figure, nonconvex uniform polyhedron, four dimensional
Topic 3	*Topology_stubs, Knot_theory_stubs, Theorems_in_topology, Theorems_in_algebraic_topology*
(Entities)	`Knot operation`, `Chromatic homotopy theory`, `Infinite loop space machine`, `Simple space`, `Base change map`
(Phrases)	topological space, algebraic topology, category theory, topological spaces, fundamental group, simply connected, homotopy theory, 3 manifold, 3 manifolds, knot theory
Topic 4	*Cryptography_stubs, Cryptography, Combinatorics_stubs, Number_stubs*
(Entities)	`PC1 cipher`, `PKCS 8`, `KR advantage`, `Ccrypt`, `BEAR and LION ciphers`
(Phrases)	dual ec drbg, block cipher, sha 1, public key, hash function, stream cipher, escape wheel, balance wheel, secret key, private key
Topic 5	*Algebra_stubs, Abstract_algebra_stubs, Linear_algebra_stubs, Theorems_in_algebra*
(Entities)	`C-closed subgroup`, `Torsion abelian group`, `Fixed-point subgroup`, `Change of rings`, `Acceptable ring`
(Phrases)	algebraic geometry, group theory, abstract algebra, finite group, finitely generated, abelian group, finite groups, galois group, commutative ring, normal subgroup

4 EXPERIMENT RESULTS

In this section, we demonstrate the effectiveness of our proposed model TaxoPhrase, by evaluating the quality of the learned topics.

Dataset. We extract the taxonomy graph, the category-page graph, and pages' content from the latest dump of the English Wikipedia[2][3]. We choose *Mathematics*[4], *Chemistry*[5], and *Argentina*[6] to construct the datasets. The resulted datasets are described in the Table 2.

Baselines and Settings. We compare our learned topics on phrases with LDA, and compare the learned topics on categories with SSN-LDA[17]. SSN-LDA utilizes the co-occurrence relation of users in the network to discover the communities. We apply it on the category-entity graph to learn the topics on categories. We set $\beta = 0.01$ for LDA and SSN-LDA, $\beta^{(\tau)} = \beta = 0.01$ for TaxoPhrase, set $\alpha = 0.1$ and do the hyper-parameter optimization every 50 sampling iterations for all methods as [15]. All the algorithms are implemented in Mallet[7] with 1000 iterations. And the topic number is set to 100 for all algorithms and datasets.

Evaluating Metric. We choose the point-wise mutual information (PMI) as the measure of the topic coherence. For each topic, the PMI are computed among all pairs of top-30 topical phrases/categories. Specifically, $PMI{-}Score(z) = \frac{1}{435}\sum_{i<j} PMI(w_{z,i}, w_{z,j}), i, j \in \{1...30\}$, where $PMI(w_{z,i}, w_{z,j})$ are computed on the reference corpus. To make the evaluation result robust, we use the whole English Wikipedia as the reference corpus for computing PMI. The final PMI is the average score over all the topics.

Effectiveness. The results are shown in Table 2. Considering the quality of the topics on phrases and categories, our proposed method TaxoPhrase both achieve the optimum scores. Also shown in the Table 1, it's easy to confirm that the joint learning on categories and phrases provide more interpretable topics. Overall, TaxoPhrase provides an effective tool to explore the knowledge base.

Table 2: The statistics of the datasets, and the evaluation result on the learned topics on phrases/categories.

		Maths	Chemistry	Argentina
#Entities		27,947	60,375	8,617
#Category Types		1,391	3,038	1,479
#Phrase Types		116,013	248,769	21,183
on phrases	LDA	4.55	4.30	3.52
	TaxoPhrase	4.67	4.55	3.81
on categories	SSN-LDA	4.01	3.97	3.06
	TaxoPhrase	4.51	4.48	3.73

5 CONCLUSION

To provide an overview information for Knowledge Base, we joint model the taxonomy structure and phrases in the entity's content. Specifically, we propose the novel model TaxoPhrase. TaxoPhrase encourages that: the category nodes in the same edge tend to share the same topic with each other; the category nodes in the same category-information tend to have very few but coherent topics; and the category nodes and the phrases are more likely to have semantically coherent topics.

The experiments on three datasets, which are *Mathematics*, *Chemistry* and *Argentina* extracted from English Wikipedia, verify the effectiveness of TaxoPhrase on exploring the Knowledge Base.

6 ACKNOWLEDGEMENTS

This research is supported by the Natural Science Foundation of China (Grant No.61572043) and National Key Research and Development Program (Project Number: 2016YFB1000704).

[2] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-categorylinks.sql.gz

[3] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

[4] https://en.wikipedia.org/wiki/Category:Mathematics

[5] https://en.wikipedia.org/wiki/Category:Chemistry

[6] https://en.wikipedia.org/wiki/Category:Argentina

[7] http://mallet.cs.umass.edu/

REFERENCES

[1] Albert-László Barabási and Eric Bonabeau. 2003. Scale-free networks. *Scientific American* 288 (2003), 50–59.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *JMLR* (2003).

[3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

[4] Jonathan Chang, Jordan L Boyd-Graber, Sean Gerrish, Chong Wang, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Nips*, Vol. 31. 1–9.

[5] Ludovic Denoyer and Patrick Gallinari. 2006. The wikipedia xml corpus. In *International Workshop of the Initiative for the Evaluation of XML Retrieval*. Springer, 12–19.

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

[7] Stefano Faralli, Giovanni Stilo, and Paola Velardi. 2015. Large Scale Homophily Analysis in Twitter Using a Twixonomy.. In *IJCAI*. 2334–2340.

[8] Lauren A Hannah and Hanna M Wallach. Summarizing topics: From word lists to phrases. In *NIPS 2014 Workshop on Modern Machine Learning and Natural Language Processing*. 1–5.

[9] Yulan He. 2016. Extracting Topical Phrases from Clinical Documents. In *AAAI*. 2957–2963.

[10] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence* 194 (2013), 28–61.

[11] Zhiting Hu, Gang Luo, Mrinmaya Sachan, Eric Xing, and Zaiqing Nie. 2016. Grounding topic models with knowledge bases. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*.

[12] Weijing Huang, Tengjiao Wang, Wei Chen, and Yazhou Wang. 2017. Category-Level Transfer Learning from Knowledge Base to Microblog Stream for Accurate Event Detection. In *International Conference on Database Systems for Advanced Applications*. Springer, 50–67.

[13] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and others. 2015. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195.

[14] Shawn Martin, W Michael Brown, and Brian N Wylie. 2007. *Dr. L: Distributed Recursive (Graph) Layout*. Technical Report. Sandia National Laboratories.

[15] Hanna M Wallach, David M Mimno, and Andrew McCallum. 2009. Rethinking LDA: Why priors matter. In *Advances in neural information processing systems*. 1973–1981.

[16] Torsten Zesch, Iryna Gurevych, and Max Mühlhäuser. 2007. Analyzing and accessing Wikipedia as a lexical semantic resource. *Data Structures for Linguistic Resources and Applications* (2007), 197–205.

[17] Haizheng Zhang, Baojun Qiu, C Lee Giles, Henry C Foley, and John Yen. 2007. An LDA-based community structure discovery approach for large-scale social networks. In *Intelligence and Security Informatics, 2007 IEEE*. IEEE, 200–207.

Multilingualization of Question Answering Using Universal Dependencies

Hiroshi Kanayama
IBM Research - Tokyo
Tokyo, Japan
hkana@jp.ibm.com

Koichi Takeda
Nagoya University
Nagoya, Japan
takedasu@i.nagoya-u.ac.jp

ABSTRACT

This paper investigates the capability of the syntax representation framework of Universal Dependencies to build multilingual applications. In a multilingual question answering system, the dependency structures are used as the input to the downstream components that can be shared for multiple languages. The experiment on the question answering pipeline demonstrates that the dependency structures commonly designed for multiple languages work better than conventional language-dependent representations, even for the Japanese language which has very different structures from those of English and Spanish.

KEYWORDS

Question answering, Dependency parsing, Universal dependencies

1 INTRODUCTION

Universal Dependencies (UD) [14] project aims to design and provide consistent treebanks for many languages, through the implementation of multilingual dependency parsers, cross-lingual transfer learning, and quantitative comparison of languages from linguistic viewpoints [10, 12]. As of the end of 2016, treebanks of 49 languages have been released [13].

Figure 1 shows the notions of data and process in typical existing studies using Universal Dependencies. First, several works tried to create UD treebanks by converting the existing treebanks of various languages, such as Russian [9], Swedish [1] and Estonian [11].

For low-resource languages, several methods of cross-lingual transfer learning have been studied, relying on richer resources in other languages, such as for part-of-speech tagging [18] and dependency parsing [5, 7, 17]. These studies were evaluated by comparing the accuracy of part-of-speech tagging and parsing with the treebanks based on Universal Dependencies.

However, there has been little work on evaluating the appropriateness of multilingual dependency representation using Universal Dependencies on multilingual downstream applications. Particularly for the Japanese language, it is still an open problem whether the Japanese dependency structures represented by Universal Dependencies are actually useful compared to conventional syntactic frameworks. Another potential issue is that the performance on the UD treebank and its usefulness for application may be different; for example, if a parser is tuned for the UD corpus, it may reduce the quality of application.

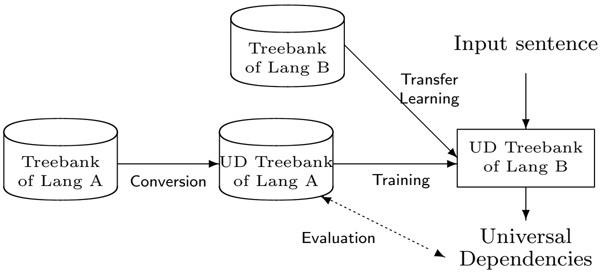


Figure 1: Typical existing studies on Universal Dependencies using languages A and B.

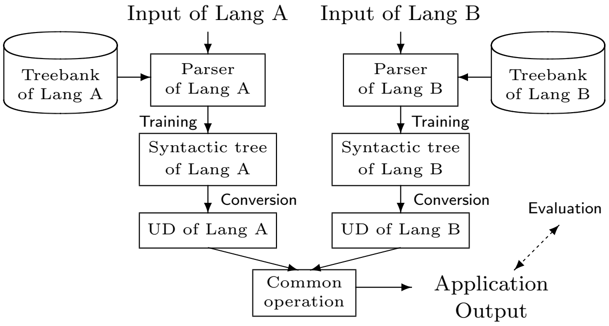


Figure 2: Concepts of a multilingual application that uses UD syntactic structure discussed here.

This paper discusses the advantage of uniform multilingual dependency structures from the viewpoint of applications, rather than evaluating the parsers of many languages themselves. As shown in Figure 2, the effect of using Universal Dependencies as a representation of syntactic structure are evaluated, by converting the multilingual dependency structure into UD representation and examining the output on a multilingual downstream component that takes the UD structure as input and applies common algorithms.

A question answering (QA) system designed for English and Spanish is used as a case study. Multiple types of representation of Japanese syntactic structures will be evaluated on this multilingual QA system, to see whether the UD can be used for a Japanese version of QA without having to implement language-specific downstream components.

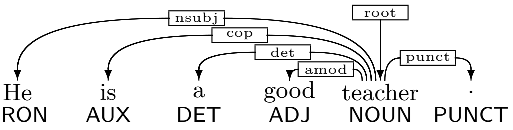
Table 1: 17 PoS tags used in Universal PoS. ∗ denotes a PoS for content words.

NOUN ∗	ADV ∗	CCONJ	PART	X ∗
PROPN ∗	PRON	SCONJ	PUNCT	
VERB ∗	NUM ∗	DET	SYM	
ADJ ∗	AUX	ADP	INTJ	

2 UNIVERSAL DEPENDENCIES

In the Universal Dependencies framework, a dependency structure is represented as in English example (1). Every word except for the root depends on another word, so a whole sentence forms a single tree.

(1)

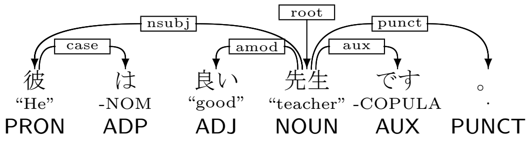


Representing only the dependencies between two words, with no regard for constituent structures, UD simplifies the tree structure, thus reducing the cost in creating treebanks. It is also robust for informal writing and ellipses.

To make the PoS system uniform across languages, the 17 PoS tags shown in Table 1 based on Google Universal Part-of-Speech Tags [15] are used. Each dependency is classified into 37 labels based on 42 labels originally defined in Universal Stanford Dependencies [4].

Rather than handling classical syntactic relationship such as agreement between a verb and its subject, UD focuses on relationships between content words in order to absorb the syntactic differences in many languages. A typical example is a copula in (1). Unlike most of the classical syntactic frameworks, which regard 'be' as the root of the sentence, and 'he' and 'teacher' as a subject and a complement of 'be' respectively, UD picks up 'teacher' as the root of the sentence, and directly connects 'he' and 'teacher'. This makes it possible to obtain a closer structure between most of the European languages with copula and languages like Russian, which do not have copula. The Japanese UD structure (2) which corresponds to (1) shows that both languages have the same root 'teacher', and two relations between content words are aligned: 'he - nsubj - teacher' and 'good - amod - teacher', even though there are differences in the PoS tags and dependency structures of functional words.

(2)



In spite of the philosophy of UD to give common representation for any language as described in Section 2, there are many open issues in UD design for Japanese [16]. For example, Japanese does not have the syntactic notion of nsubj, obj, iobj, so it is not easy to attach those labels to the arguments of a verb and an adjective. Also it is difficult to draw a line between acl and amod because the attributive adjective behaves like a relative clause in Japanese.

The next two sections discuss how to apply the Japanese UD structure to the common downstream components without having to worry about intrinsic inconsistency of syntactic structures.

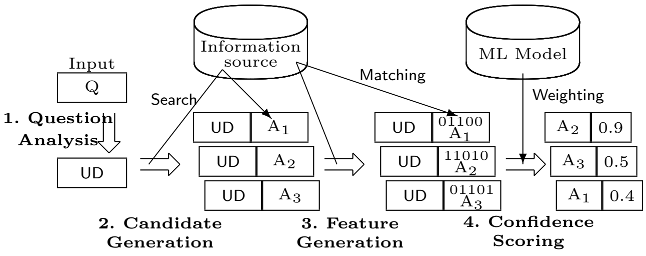


Figure 3: The flow of multilingual factoid QA.

3 MULTILINGUALIZATION OF A QA SYSTEM

As a case study of UD application, a multilingual question answering system is adapted for an additional language. QA is selected because it is one of applications that benefit from multilingual information sources. The open domain factoid question answering system for English, DeepQA [6], has been redesigned to accept Spanish and other European languages [3] based on Universal Dependencies as common syntactic representation.

Figure 3 shows the flow of question answering discussed in this paper. To realize its multilingualization, language dependent operations are consolidated into the question analysis part, and the downstream components will be designed for many languages. Here is the simplified pipeline for multilingual QA:

1. **Question analysis** parses the input question and convert the parse tree into the UD structure (denoted as UD in Figure 3), and then obtains the type of the answer.
2. **Candidate answer generation** searches on the documents stored in the information source using the words extracted from the input question as the query, and then enumerates the titles of the documents and anchor links in the documents as candidate answers. The search query is generated by enumerating the content words (see Table 1) in the UD tree. Using Wikipedia as the information source, it is possible to use the common logic for this component since it has uniform structures for any language.
3. **Feature generation** calculates the multiple similarity values between the information source and question filled by each candidate answer, referring to the passages obtained by secondary search from the information source. Those values are then used as features of each candidate answer. For the calculation, PoS tags

and relation labels of UD structures are used to detect content words and phrases.

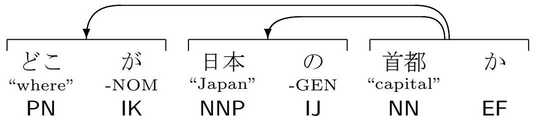
4. **Confidence scoring** applies logistic regression to weight the features generated in the previous component, using the training data consisting of pairs of a question and correct and wrong answers. Then the confidence value is calculated for each candidate answer as the inner product of a feature vector and weights of a feature, and the candidate answer with highest confidence will be selected as the output of the system. This process is completely language independent.

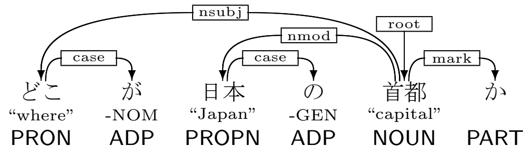
This approach has enabled the question answering for English and Spanish. Now the question is whether the same approach is valid for Japanese which has very different syntactic structures and units of words. Section 4 discusses the capability of multilingualization using the UD framework.

4 USAGE OF JAPANESE UD

4.1 Conversion of Parse Tree

To obtain a dependency structure compatible with the Japanese UD definition [16], we convert the phrase-level output of the Japanese syntactic parser [8] into the word-level dependency structures. For instance, a dependency structure (3) ("What is the capital of Japan?") is converted into the UD format (4).

(3)


(4)


The UD structure is used as the output of the question analysis component of the QA system in Figure 3, and it is consumed in the downstream multilingual components.

4.2 Experiment

To determine the effectiveness of the UD-compliant structure as an input to multilingual components, we ran the whole pipeline of the QA system with varying dependency structures. For the evaluation and training, an existing set of open domain factoid questions were translated into Japanese as in Table 2. Japanese Wikipedia articles were used as the information source. For simplicity, language dependent features have been removed from the feature generation part, though there may be improved quality for each language.

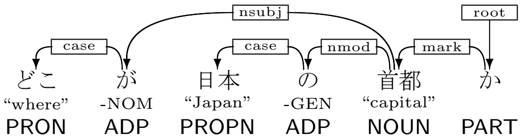
As the output of the question analysis component, we considered, by artificial conversion of labels and dependency structures, the following syntactic structures.

Table 3: QA performance with 220 test questions.

	Recall	Accuracy
(A) UD Compliant	67.3%	15.0%
(B) Conventional dependency	62.7%	10.5%
(C) Without relation labels	62.7%	10.0%
(D) Without PoS tags	58.6%	14.5%
(E) Randomized PoS	34.1%	2.7%
(F) Search only	41.3%	3.6%

(A) UD compliant. A syntactic structure compatible with UD definition as exemplified by (4). It is converted from the original Japanese parsing structure as (3).

(B) Conventional dependency structure. A word-level dependency structure in which all dependencies have right-head direction as (5). UD-style labels are assigned to relations, though some of them have opposite dependency direction from the UD definition.

(5)


(C) Without relation labels. Use only dep (default value) as the relation labels for all dependencies.

(D) Without PoS tags. Use only X (default value) as the PoS tags for all words.

(E) Randomized PoS. Randomly assign the 17 PoS tags in UD definition.

Table 3 shows recall (the ratio at which the correct answer appeared in the top 100 candidates) and accuracy (the ratio at which the answer with the highest confidence value was correct) in the QA system, with using (A) to (E) above in the question analysis component. 'Search only' is the baseline method that naively searches the Wikipedia articles and outputs the title of the most relevant document. Its low accuracy indicates that there are few questions that can be solved trivially. The following discussion focuses on relative performance among various syntactic representations rather than absolute value of the quality, since it highly depends on the complexity of the questions and coverage of the information source.

By converting into the common syntax structure as (A), the whole system worked well enough without implementing any language-specific components in the pipeline. When a different form of dependency structures was used as (B), or when relation labels were missing as (C), recall was decreased because the selection or weighting of the words and phrases for search query were not optimized. Also the accuracy became lower in (B) and (C) because the coincidence of dependencies between two content words in the feature generation was not captured [1].

[1] For example, the relationship between 'Japan' and 'capital' can be obtained in (4), but not in (5).

Table 2: Example of questions and answers.

en	Which country was admitted into the World Trade Organization in August 2012?	Russia (Vanuatu)
ja	2012 年 8 月に世界貿易機関への加盟が承認された国はどこか？	ロシア (バヌアツ)

When all PoS tags were replaced by X in (D), all words were regarded as content words and the recall was reduced due to the noises in the query, but as long as the correct dependency structures were captured, the correct answer could obtain a higher confidence value, so the loss of the accuracy was limited. When the PoS tag was randomized, the content words to build the search query were not correctly obtained, so both the recall and accuracy were drastically reduced.

5 CONCLUSION

This paper examined the contribution of Universal Dependencies to the design of multilingual application. Simply by providing UD based syntactic structures in each language, whole QA pipeline worked, since the downstream component was appropriately generalized to use the syntactic structure to generate search queries and to compare the question and search results within the language.

In this study only language-independent features are used with separated information source by languages. By combining deeper common structure such as universal semantic role label [2], the QA is expected to be enhanced using cross-lingual information source.

If more applications to be evaluated on multiple languages are identified the effectiveness of the universal syntactic structure can be estimated quantitatively. This will enormously help the design of Universal Dependencies, which will be of great benefit to multilingual applications.

REFERENCES

[1] Lars Ahrenberg. 2015. Converting an English-Swedish Parallel Treebank to Universal Dependencies. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*. 10–19.

[2] Alan Akbik and Yunyao Li. 2016. Polyglot: Multilingual semantic role labeling with unified labels. In *Proceedings of ACL 2016*.

[3] Keith Cortis, Urvesh Bhowan, Ronan Mac an tSaoir, D.J. McCloskey, Mikhail Sogrin, and Ross Cadogan. 2014. What or Who is Multilingual Watson?. In *Proceedings of COLING 2014: System Demonstrations*. 95–99.

[4] Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford Dependencies: A cross-linguistic typology. In *Proceedings of LREC*. 4585–4592.

[5] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. 845–850.

[6] D. A. Ferrucci. 2012. Introduction to "This is Watson". *IBM Journal of Research and Development* 56, 3.4 (2012), 1:1–1:15.

[7] Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2016. A Universal Framework for Inductive Transfer Parsing across Multi-typed Treebanks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 12–22.

[8] Hiroshi Kanayama, Kentaro Torisawa, Yutaka Mitsuishi, and Jun'ichi Tsujii. 2000. A Hybrid Japanese Parser with Hand-crafted Grammar and Statistics. In *Proceedings of the 18th International Conference on Computational Linguistics*. 411–417.

[9] Janna Lipenkova and Milan Souček. 2014. Converting Russian Dependency Treebank to Stanford Typed Dependencies Representation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics(EACL)*. 143–147.

[10] Ryan T McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal Dependency Annotation for Multilingual Parsing.. In *ACL (2)*. 92–97.

[11] Kadri Muischnek, Kaili Müürisep, and Tiina Puolakainen. 2016. Estonian Dependency Treebank: from Constraint Grammar tagset to Universal Dependencies. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France.

[12] Joakim Nivre. 2015. Towards a Universal Grammar for Natural Language Processing. In *Computational Linguistics and Intelligent Text Processing*. Springer, 3–16.

[13] Joakim Nivre, Željko Agić, Lars Ahrenberg, et al. 2017. Universal Dependencies 2.0 — CoNLL 2017 Shared Task Development and Test Data. (2017). http://hdl.handle.net/11234/1-2184 LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

[14] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, 1659–1666.

[15] Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of LREC*.

[16] Takaaki Tanaka, Yusuke Miyao, Masayuki Asahara, Sumire Uematsu, Hiroshi Kanayama, Shinsuke Mori, and Yuji Matsumoto. 2016. Universal Dependencies for Japanese. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016*.

[17] Jörg Tiedemann. 2015. Cross-Lingual Dependency Parsing with Universal Dependencies and Predicted PoS Labels. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*. 340–349.

[18] Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham, and François Yvon. 2014. Cross-Lingual Part-of-Speech Tagging through Ambiguous Learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Wikipedia Based Essay Question Answering System for University Entrance Examination

Takaaki Matsumoto
Carnegie Mellon University
5404 GHC 5000 Forbes Ave.
Pittsburgh, Pennsylvania, USA 15213
SOC Corporation
3-16-17, Takanawa, Minato-ku
Tokyo, Japan 108-0074
takaaki@gmail.com

Francesco Ciannella
Fadi Botros
Evan Chan
Cheng-Ta Chung
Keyang Xu
Tian Tian
fciannella@cmu.edu
fbotros@andrew.cmu.edu
yiksanc@andrew.cmu.edu
chengtac@cs.cmu.edu
keyangx@andrew.cmu.edu
tian.tian@cmu.edu
Carnegie Mellon University
5404 GHC 5000 Forbes Ave.
Pittsburgh, Pennsylvania, USA 15213

Teruko Mitamura
Carnegie Mellon University
6711 GHC 5000 Forbes Ave.
Pittsburgh, Pennsylvania, USA 15213
teruko@cs.cmu.edu

ABSTRACT

This paper describes an open knowledge (Wikipedia) based question answering system that generates essays to answer the real examination questions for the admission to the Tokyo University. Questions are formulated in English and their answers are also expected in English, although they are to be found in Japanese language textbooks. This cross-lingual narrow domain question task is a hard task because most questions are based on the limited target language knowledge base which is only available in its original language. Large scale open-domain knowledge resources will certainly contain the answers, but retrieving them is difficult due to their inherent high signal to noise ratio. To overcome Wikipedia's high signal to noise ratio, we carefully calculate the weights of the keywords extracted from the question, based on a tf-idf score of the entire Wikipedia. The relevant articles are then retrieved and sets of passages are extracted based on the weighted keywords. Cherry picking, generative method, or sentence ordering strategies are subsequently used to generate short or long essays. The results of the end-to-end evaluation indicate that the proposed system succeeded to generate better essays compared with the previous research that also uses Wikipedia and the reference system that uses machine translated Japanese textbooks.

KEYWORDS

Question answering, open knowledge base, summarization, NTCIR-13, world history

1 INTRODUCTION

Question answering (QA) is one of the most notable natural language processing applications and has been heavily researched for several decades. While most research focuses on factoid, true/false and multiple choice QA tasks, essay QA has been proven to be one of the more challenging tasks since it usually requires a deeper understanding of the subject matter, information extraction from multiple sources and summarization to produce a coherent essay.

NTCIR (NII Testbeds and Community for Information access Research) [1] is a series of workshops that expand research in Information Access (IA) technologies including information retrieval, question answering, text summarization, extraction, etc. QA Lab [2], one of the tasks of NTCIR, aims to investigate complex real-world QA technologies as a joint effort of participants. The QA tasks of the NTCIR 13 QA Lab consist of three type of questions: multiple choice, named-entity and essay type questions from Japanese university entrance examination, which focus on world history [18][17].

In this paper, we present our system which participated in the essay QA portion of QA Lab 3. The rest of the paper is layed out as follows. We further explain the task in section 2. In section 3, we discuss the design of the system in detail and show evaluation for each module. Finally, in section 4 we present end-to-end system evaluation.

2 TASK AND REFERENCE SYSTEM

The essay QA task of NTCIR QA Lab 3 contains short/simple and long/complex essays. The former requires an answer essay of one or two sentences (from 15 to 60 words) with some of them containing a factoid type question. The latter expects multiple (usually from 5 to 8) sentences (225-270 words), and has 8-10 keywords that should be used in the essay. Examples of the questions are following:

Short essay The Inca Empire had no writing system, but it controlled the large territory of the Andes. Describe, in 15 English words, the transportation and information methods used by the Empire.

Long essay In answer space (A), in 225 English words or less, describe the historical significance of the philosophies of these intellectuals, including the conditions in the 18th century which led them to these conclusions, especially in France

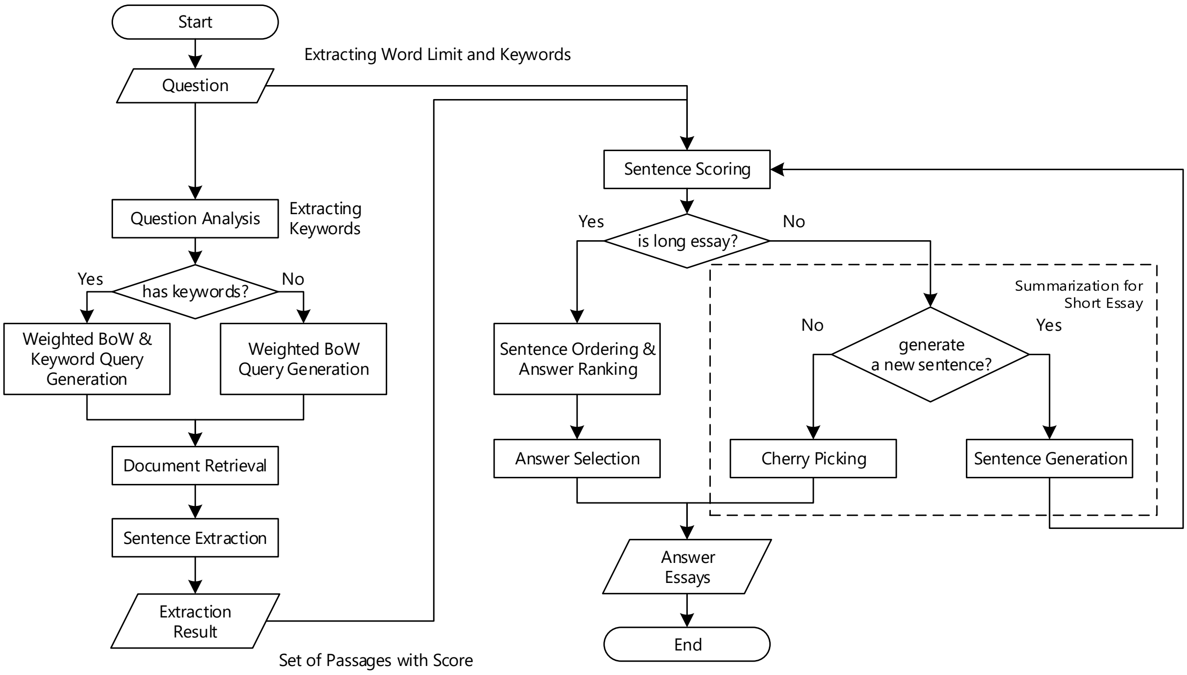


Figure 1: System Flowchart.

and China. Use each of the terms below once, and underline each term when it is used; Society of Jesus, imperial examinations, enlightenment, absolute monarchy, revocation of the Edict of Nantes, French Revolution, class system, Literary Inquisition.

A multilingual essay question answering system developed by Sakamoto's et al. [9][16] has been employed as the reference system. Knowledge resources for the reference system are five machine translated (Google Translate, in 2015) Japanese world history textbooks.

3 SYSTEM MODULES AND THEIR EVALUATIONS

3.1 Overview

Fig. 1 shows the architecture of the end-to-end system. Detailed architecture, algorithms used and experimental design for each of these modules are covered in detail in the following subsections. Communication between each module is performed using JSON files for ease of use and readability. To ensure consistency between each test iteration, we run each end to end test using a build automation software called Jenkins. Whenever a new JSON file is produced and committed to the git repository by the extraction subsystem, the build automation mechanism detects the changes and triggers the start of the summarization system which ultimately produces the results of the evaluation in an HTML report that can be consulted on-line.

The main data source we used to extract answers is Wikipedia. We experiment with a dump of all of Wikipedia and only the history section of Wikipedia [8].

3.2 Question Analysis

Question analysis module is meant to extract all useful information from question data to serve all remaining modules in our end-to-end system. This module is composed of three components, namely, information extraction, text processing and weighted keywords generation.

Information extraction refers to extracting values of a few XML tags (e.g., <instruction></instruction>) which helps solving question answering problems from three sources: qalab3-en-phase1-answersheet-essay.xml, qalab3-en-phase1-essay-extraction-GSN.xml and qalab3-en-phase1-goldstandard-essay.xml.

Extraction is followed by text processing. In NTCIR questions, redundancy for information retrieval exists and all these patterns are needed to be removed otherwise they add noise to information retrieval, e.g., "Write your answers in the answer space".

Weighted keywords generation means generating a list of reasonable keywords with corresponding weights from question text (Note that long essay questions provide a list of keywords). There are multiple ways to generate keywords and assign them different weights. After a series of experiments, we use Tf-idf metrics as the weight generator for two reasons. First of all, the algorithm make few assumption on the data. Secondly, it is properly implemented in scikit-learn.

To calculate tf-idf, we append all 27 question text (i.e., concatenated by instruction, grand_question and reference field) from XML source file to History Wikipedia corpus consisting of 11217 documents, and construct a new corpus with 11244 (=11217+27) documents. Then tf-idf weights on the corpus are calculated, and all phrases in 27 questions are sorted by tf-idf value. After that, for those question with given keywords, append these keywords into keyword list and assign them weight of 1 (heuristically). As a result, those phrases with highest tf-idf values are labeled as keywords, and be sent to information retrieval module along with their tf-idf value as weight.

3.3 Document Retrieval

The document retrieval module indexes information from Wikipedia and retrieves relevant text records against structured queries generated based on questions. The Wikipedia history subset created by Wang et al. [20] was used as the collection for constructing the index. Indri [19] was utilized for indexing, which was stopped using the default Indri stoplist and stemmed using the Krovetz stemmer.

Each Wikipedia page can be indexed as a document, which is the basic unit for retrieval. This is known as the page-level indexing. However, the question answer requires to locate the exact paragraph in extracting specific sentences relevant to the question and the whole Wikipedia page might contain too much noise. In order to increase the accuracy of sentence extraction, we also adopted passage-level indexing; It divided each Wikipedia page into sentences using Standford CoreNLP tool [1] and used a sliding window approach to combine sentences into passages [4]. Here, the sliding windows contains 10 sentences without overlapping and we heuristically chose 10 because the question answer is required to have around 40-60 words.

Structured queries are generated with weighted keywords extracted from the Question Analysis module(see Section 3.2). An example for keywords set “Olympia; Greek; 4th century CE ” is shown as follows:

$$\#combine(\alpha_1\ Olympia\ \alpha_3\ Greek\ \alpha_3\ \#1(4th\ century\ CE))$$

where α_1,α_2,α_3 are weights generated by the Question Analysis module; And #1() operator requires all terms inside appear continuously.

The retrieval model is Indri [14], which combines statistical language models and Bayesian inference networks. All parameters were default settings. Top 20 retrieved documents are ranked with Indri scores and returned for the Sentence Extraction module in the next step.

3.4 Sentence Extraction

This module takes the output of the question analyzer and document retrieval modules and extracts sentences that could be potential answers to the question. It uses the original question, the list of retrieved documents and attempts to extract sentences that contain the answer to the question. Since long and short essay questions have different answering requirements, the system uses different strategies to answer them. This module consists of following submodules:

Document Cleaning Raw Wikipedia files are highly noisy: they contain a lot of tables, links, citations, markup, etc. That is why it is important to clean the documents and remove all the unnecessary content. This sub-module also segments the documents into sentences and tokenizes each sentence. These are the steps that are taken to process each document:

1. Remove documents that do not actually contain any useful text but rather contain a list of links to other pages (e.g. Category pages)
2. Segment documents into sentences
3. Filter out sentences that:
   - (a) Contain links
   - (b) Contain HTML or Wikipedia markup
   - (c) Are image captions
4. Tokenize each sentence:
   - (a) Remove non-alphanumeric characters
   - (b) Remove stopwords
   - (c) Lowercase tokens
   - (d) Stem tokens
5. Remove sentences that contain two tokens or less
6. Remove duplicate sentences

Passage Extraction There are separate passage extraction submodules for short and long essays since different strategies are used to answer each type of question. Each sub-module takes in the output of the question analyzer and the cleaned documents and outputs a list of sentences that are potential answers to the question. The algorithms used by these sub-modules are outlined in detail in the following section.

Evaluation The evaluation sub-module uses the given gold passages to evaluate the extracted passages. It uses both human annotations and automated methods to evaluate the performance of passage extraction. It also contains scripts that attempt to make human annotation of extractions as fast and efficient as possible.

3.4.1 Algorithms. Following algorithms were tested to select the most suitable algorithms for sentence extraction.

Jaccard Similarity Similarity is calculated between all the words in the question (introduction/instruction paragraphs and given keywords) and each sentence from the retrieved documents then the top 10 sentences with the highest scores are chosen.

Field-weighted Jaccard Similarity : Since the introduction paragraph is usually longer than the instruction paragraph, the sentences extracted using Jaccard similarity tended to be more relevant to the introduction paragraph but not to the actual instruction paragraph. Therefore, to remedy this problem, the following formula was used to give more weight to the instruction paragraph:

$$\begin{aligned}\text{Score(Question, Sentence)} &= \\ &0.7 * \text{Jaccard(Instruction, Sentence)} \\ &+ 0.3 * \text{Jaccard(Introduction, Sentence)}\end{aligned} \tag{1}$$

Field-weighted Jaccard + MMR Wikipedia contains many sentences that are very similar to each other terms of content. Therefore, sometimes the system would return 10 sentences that are all very similar. This is not very beneficial for this

[1] https://stanfordnlp.github.io/CoreNLP/

task, especially for long essay questions where we want to cover a wide range of topics. To diversify the extracted sentences, MMR is used. The essence of MMR [5], which is a greedy algorithm, is in each iteration, it would pick passage that has high relevant score with question but also with little overlap with selected passages.

Field-weighted TF-IDF History questions tend to contain many names of people, events, places and special words that should be given more weight since the question is usually focused on those words. Therefore, TF-IDF and cosine similarity are used to rank sentences. IDF values are calculated using the entire Wikipedia corpus.

Field-weighted TF-IDF + PM2 Long essay questions contain keywords that have to be used and discussed in the essay. However, the previous methods cannot guarantee that all keywords were covered in the extracted passages. It is possible that all the extracted passages are only relevant to one keyword (or none at all). The PM2 diversification algorithm [6] is used to try to increase keyword coverage for long essay questions. PM2 is generally used in document retrieval when the query can have multiple intents and we want to retrieve documents that address all the intents proportionally. It gives a higher score to documents that cover multiple intents. Similarity, in long essay questions we want to retrieve sentences that cover each keyword proportionally and give extra weight to sentences that cover multiple keywords. Sentences that cover multiple keywords can connect the given concepts and potentially produce a more coherent essay.

3.4.2 Evaluation. We focused on human annotations to evaluate the extracted passages. A binary relevance metric was used to evaluate each extracted passage and precision @10 and mean reciprocal rank were then calculated for each experiment. For long essay questions, keyword recall is also evaluated by measuring the fraction of keywords that are present in the extracted passages.

Table 1 summarizes the results for each of the tested algorithms. The results show that field-weighted TF-IDF + PM2 gave the best results for all metrics.

As mentioned above, using simple Jaccard similarity is naive since most of the extracted passages were relevant to the introduction paragraph but not to the actual question. Using field-weighted Jaccard doubled the precision scores which indicates that it is effective. While MMR reduced redundancy, the results show that it didn't improve precision or MRR thus it is questionable whether it is useful or not for this task. As predicted, TF-IDF was a very effective method to improve results as demonstrated by the improved precision and MRR. However, TF-IDF gave the lowest keyword recall for long essays but PM2 proved effective as it doubled keyword recall while also improving precision/MRR for long essays.

3.5 Sentence Scoring

The sentence scoring module gives a score to the extracted set of the passages. Since the questions are entrance examination, they need the existence of important keywords in the essay. Therefore, the simplest sentence scoring methods is measuring keyword entailment.

Table 1: Evaluation Result of Passage Extraction Algorithms

Algorithm	Short Essays		Long Essays		
	P@10	MRR	P@10	MRR	Keyword Recall
Jaccard	0.077	0.330	0.520	0.850	0.447
Field-weighted Jaccard	0.150	0.432	0.700	0.767	0.509
Field-weighted Jaccard + MMR	0.109	0.444	0.660	0.733	0.529
Field-weighted TF-IDF	0.191	0.447	0.679	0.750	0.376
Field-weighted TF-IDF + PM2	0.191	0.447	0.720	0.900	0.714

$$\text{Score} = \frac{k_s}{m} \tag{2}$$

where k_s is the number of keywords in the sentence, and m is the number of words of the sentence. All keywords and words of the sentence are stemmed. Stop words and punctuations are removed before calculation.

Eq.2 measures the density of the keywords in a sentence. However, not always the given keywords and words in the sentence exact match. Some words of the answer sentence could be similar to the given keywords. Hence, word level similarity between retrieved or given keywords and an extracted sentence is calculated as follows:

$$\text{Score1} = \sum_{i=1}^{m} \frac{\max(w_i \cdot k_1, w_i \cdot k_2, ...w_i \cdot k_n)}{m} \tag{3}$$

where, m is the number of words in the sentence, n is the number of keywords, w is the word vector, k is the keyword vector. Word embedding is given by GloVe (6B 100d) [13].

Eq.3 calculates similarity between given keywords and all words in an extracted passages. With this scoring method, the mean of the ROUGE-1, which is one of the official answer scoring methods of the NTCIR QA Lab [17], are improved (from 0.0598 to 0.0671) in the phase-1 data. However, dividing by the sentence length means measurement of the similarity density of a passage. In general, the longer sentence, the more information exists. Hence, we can modify the formula as follows:

$$\text{Score2} = \sum_{i=1}^{m} \frac{\max(w_i \cdot k_1, w_i \cdot k_2, ...w_i \cdot k_n)}{\log m} \tag{4}$$

The objective of the division by logarithm of sentence length is to consider the information density and amount simultaneously. The ROUGE-1 mean improved compared with the previous formula (from 0.0671 to 0.0680).

Above sentence scoring methods are keyword based (word level) approach. Today it is not difficult to calculate sentence embedding vector. Assuming that an entailment exists between questions and answers, sentence score can be given as following:

$$\text{Score3} = \max(\text{sim}(s, q_1), \text{sim}(s, q_2), ..., \text{sim}(s, q_l)) \tag{5}$$

where, sim is the function to calculate sentence similarity between two sentences, s is the extracted sentence, q_i is the i-th sentence of the question, and l is the number of sentences of the question. Sentence similarity is calculated by a siamese Long Short-Term Memory (LSTM) [12]. The siamese LSTM is one of the state of the art to assess semantic similarity between sentences. It uses word-embedding vectors supplemented with synonymic information to the LSTMs, which outputs a fixed size vector to encode the meaning expressed in a sentence. By calculating simple Manhattan metric, it gives the sentence representations to form a space which reflects semantic relationships.

3.6 Text Ordering for Long Essay

Answer candidates for long essays are generated by this module. This module has two models. The first one is K-Means model, which tries to capture the relation between sentences to generate coherent essay. The other one is MMR model, which does not aim at coherent essay. Instead, it tries to diversify the topics to generate the essay.

3.6.1 K-Means model. In [21], Zhang proposed summary generation by using global and local coherence. The intuition of this model is that there are 2 kinds of coherence: global coherence and local coherence. The global coherence means the connectivity between remote sentences. It is more like sub-topic transition, for example usually essay would cover "cause" of events first, then the "result" of events. On the other hand, local coherence indicates the connectivity between adjacent sentences, such as using some transition words to connect two sentences. Because coherence can be regarded as some kind of similarity between sentences, thus, this module adopts cosine similarity to measure the coherence.

To capture the coherence, this module applies K-means in scikit-learn package [3] to cluster the input passages. this module assumes that each cluster is related to different sub-topics, as the similarity within each sub-topics should be very similar. Each passage is represented as a word vector, whose value is tf-idf of the words in each dimension.

After the clustering, the next step would be to generate the order of these clusters, the sequence of sub-topics, to achieve global coherence. To do this, the system greedily pick most coherent cluster with ordered clusters. For local coherence, the strategy is similar that the system would greedily pick passage from the cluster with maximum coherence with selected sentences.

3.6.2 MMR Model. For this model, the idea is that while K-Means model may generate coherent sentence sequence, the gold standard essay is not usually coherent because it has not only to cover all specified keywords but also to fulfill the words length constraint as well. Therefore, it may be useful to select sentences that cover keywords from different aspects but also be relevant to the question. Another reason is that although MMR may not be able to generate coherent essay, the evaluation metric does not consider the coherence either. Thus, it would still be beneficial if the system can select good candidate sentences.

3.6.3 Evaluation. The dataset for the evaluate of this long essay generation module are the gold standard passages and gold standard essays provided by NTCIR. There are 5 long essay questions, and each of them is associated with several passages and 3 gold standard essays. In this evaluation, the gold standard passages are used as input to the system, and gold standard essays are used to evaluate the essay generation system. The evaluation metrics is ROUGE-1 and ROUGE-2 means[10].

Table 2 shows the performance for these 2 models, and different parameter K for K-Means models. The combined method is to pick an essay generated from above methods that has highest relevant score with question. We can see that ROUGE-1 score is the same for all K-Means methods, it is because the sentence removal strategy would remove almost the same sentences, and ROUGE-1 only measures on single words. For ROUGE-2, the score is different as it measures on bi-gram, it improves when K grows from 1 to 3, then decreases gradually after that. It indicates that the clustering is effective, while the number of clusters should not be too large, as there are generally around 7 to 9 sentences in the gold standard essays.

Table 2: Results of long essay models evaluation

Method	ROUGE-1	ROUGE-2
K-Means (K=1)	0.584	0.356
K-Means (K=3)	0.584	0.358
K-Means (K=5)	0.584	0.357
K-Means (K=7)	0.584	0.352
MMR	0.596	0.396
Combined	0.596	0.396

3.7 Summarization for Short Essay

The Summarization for short essay module provides a way to summarize a set of sentences coming from the upper layers to produce a fixed length short essay, following the directions provided in the question. The summarization paradigm that has been used is the abstractive summarization, which tries to leverage on the semantics of the sentences to achieve the text compression. The module uses three possible summarization techniques, returning only the best result to the evaluation module. Two of the techniques are actually pure abstractive summarization techniques, the third one is a trivial NLP based summarization technique. The first two techniques are implementations of the two main research approaches in abstractive summarization: AMR graph merging and Neural Network with attentional model.

3.7.1 AMR model [11]. Abstractive summarization is one of the hard NLP tasks that is still an open field of research with very few techniques, unlike other NLP tasks. It it a task that cannot be decoupled from semantics: to be able to create an abstract summary of a passage, one needs to have a deep insight into what is the meaning that the passage bears. Therefore we thought to use AMR, which is one of the resources available in NLP for implementing semantics. A thorough description of the algorithms that we used can be found in [11], and we remind the reader to that paper for the details. We implemented the algorithms described in that paper, and on top of it we laid down the basis to add to the pipeline the generative model (in the paper the generation of the summary from the summarized AMR graph is left to a mere bag of words). The generative model is able to create a well formed sentence from an AMR graph (with the limitations of AMR, like for instance the impossibility of using verbs tenses).

AMR provides a whole-sentence semantic representation, represented as a rooted, directed, acyclic graph. Nodes of an AMR graph are labeled with concepts, and edges are labeled with relations. Concepts can be English words, PropBank event predicates, or special keywords. The core semantic relations are adopted from the PropBank annotations; other semantic relations include "location," "mode," "name," "time," and "topic."

In the AMR summarization framework, summarization consists of three steps

(1) parsing the input sentences to individual AMR graphs,
(2) combining and transforming those graphs into a single summary AMR graph
(3) generating text from the summary graph.

The graph summarizer, first merges AMR graphs for each input sentence through a concept merging step, in which coreferent nodes of the graphs are merged; a sentence conjunction step, which connects the root of each sentence's AMR graph to a dummy "ROOT" node; and an optional graph expansion step, where additional edges are added to create a fully dense graph on the sentence level. These steps result in a single connected source graph. A subset of the nodes and arcs from the source graph are then selected for inclusion in the summary graph. Ideally this is a condensed representation of the most salient semantic content from the source.

We used the proxy report section of the AMR Bank because a dataset for a summarization task should include inputs and their summaries, each with gold-standard AMR annotations. A proxy report is created by annotators based on a single newswire article, selected from the English Gigaword corpus.

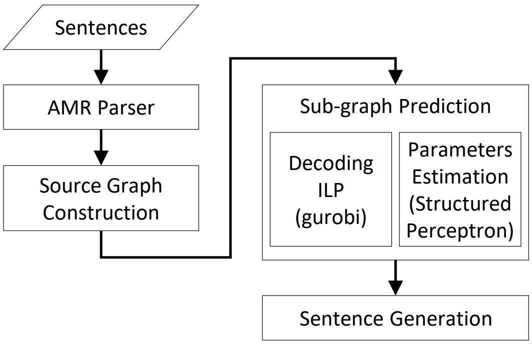


Figure 2: AMR Model Architecture

Fig. 2 shows the architecture of the AMR model. The summaries of components are following:

Source Graph Construction The "source graph" is a single graph constructed using the individual sentences' AMR graphs by merging identical concepts. Concept merging involves collapsing certain graph fragments into a single concept, then merging all concepts that have the same label.
Ideally, a source graph should cover all of the gold standard edges, so that summarization can be accomplished by selecting a subgraph of the source graph

Subgraph Prediction This steps selects a summary subgraph from the source graph. This is done with a structured prediction algorithm that enforces the following constraints in the statistical model for subgraph selection: include information without altering the meaning, maintain brevity, and produce fluent language. The selection of the graphs is done using ILP (Integer Linear Programming)

Decoding Decoding is performed as an ILP task with the constraints that the output forms a connected subcomponent of the source graph.
The length constraint is used to fix the size of the summary graph (measured by the number of edges). This is an important parameter in that the performance of a summarization systems depends strongly on their compression rate, and it is important for the NTCIR purpose because of the length limitations on the essays. An exact ILP solver called Gurobi is used.

Parameter Estimation Source graphs and summary graphs, represent a collection of input and output pairs, therefore we can use a Machine Learning algorithm like the structured perceptron to learn the parameters of the objective function designed in the previous set.

Generation Generation is the weakest link in the current chain. At the moment it is no more than a bag of words, but the plan is to plug it into a language generator from AMR.

3.7.2 Neural model with attention [15]. The idea of using a neural attentional model for summarization comes after the recent success of neural machine translation. The idea is to combine a neural language model with a contextual input encoder that learns a latent soft alignment over the input text to help inform the summary. Both the encoder and the generation model are trained jointly on the sentence summarization task.

Given an input sentence, the goal is to produce a condensed summary. The key takeaway is that the abstractive summarization task can be formulated mathematically as in finding the output sequence $\boldsymbol{y}$ that maximizes a scoring function over the input sequence $\boldsymbol{x}$ and $\boldsymbol{y}$ itself.

The problem boils down in modeling the following probability:

$$p\left(\boldsymbol{y}_{i+1}|\boldsymbol{x}, \boldsymbol{y}_c; \theta\right) \tag{6}$$

where $\boldsymbol{y}_c$ is a window of size c over the previous tokens in the output sequence and θ is the parameter of the neural network. Instead of using a noisy-channel approach, the original distribution is directly parametrized as a neural network. The network contains both a neural probabilistic language model and an encoder which acts as a conditional summarization model.

The attentional based model can be regarded as a model that learns a soft alignment, P, between the input and the summary.

Once we have the augmented language model, the generation of a summary is a search problem over a scoring function:

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y} \in Y} \sum_{i=0}^{N-1} g\left(\boldsymbol{y}_{i+1}, \boldsymbol{x}, \boldsymbol{y}_c\right) \tag{7}$$

where N is the length of the output sentence and g is the scoring function. This is the decoding problem that can be accomplished using beam search.

The training dataset is the Gigaword dataset. The golden standard summary is the headline of the news article and the body of the document is represented by the first two sentences in the article.

3.7.3 Pick one sentence. The pick one sentence method simply selects one sentence among the candidate ones, based on the relevance score provided by the scoring system and by the closeness to the required sentence's length.

This model was also designed to implement some basic NLP features, like for instance providing the sentence up to a punctuation mark or by pruning the lexical pare tree of the sentence ad hoc but so far we figured that the simple greedy method worked better in terms of evaluation scores.

4 END-TO-END EVALUATION

Table 3 shows the settings of the proposed system. The combination of "sentence similarity scoring (SentSimScoring)" and "generative short essay (Generative)" was not attempted because of both algorithm takes very long time.

Table 4 shows the summary of our system result. NTCIR employs machine evaluation and human experts to score essays. The ground truth essays are provided from the NTCIR official data. Since ROUGE-1 and 2 [10] are used one of the evaluation methods of the NTCIR QA Lab[17], ROUGE-1 and 2 scores of the proposed system were calculated using the evaluation function of the reference system [16].

Comparing the all systems in the Table 4, the Wiki-WordSimScore-PickOne has the best end-to-end ROUGE-1 mean. In addition, even though it should be noted that the results of the proposed system and the previous research cannot be simply compared because of the different questions, the ROUGE-1 mean score of the answers generated by Wiki-ExtractionScore-PickOne is about four times larger than that of the previous study that also uses Wikipedia (0.0326 in ROUGE-1 mean) [7].

The previous research by Day et al. [7] is the only available result for the NTCIR QA Lab essay QA task. The reference system (FelisCatusZero) developed by Sakamoto et al. is also the only open source software for the NTCIR QA Lab task. Compared with these two studies, the proposed system achieved high ROUGE-1 mean for the NTCIR QA Lab 3 phase-1 data. However, it also should be noted that the number of the question is only a few (5 long essays and 22 short essays). Since the NTCIR QA Lab uses the real past entrance examination of University of Tokyo, the provided data was very small. Considering the standard deviations in the table 4, the performance differences are not statistically significant.

Table 5 shows the comparison of end-to-end, short and long essay task ROUGE-1 and 2 means. It indicates that most of the ROUGE-1 and 2 mean progress comes from the short essays. Generative algorithm for short essay, Wiki-WordSimScore-Generative, was relatively worse than cherry picking (Wiki-WordSimScore-PickOne) for short essay task, however, in some questions the generative model worked better than the cherry picking.

As for the long essay question, the ROUGE-1 means of all four end-to-end conditions (FelisCatusZero, Wiki-ExtractionScore-PickOne, Wiki-WordSimScore-PickOne, and Wiki-SentSimScore-PickOne) are approx. 0.2. These results indicates that the effectiveness of the sentence scoring methods are almost the same, even if their methodologies are different. However, the ROUGE-1 mean of GSN-WordSimScore-PickOne which used the gold standard extraction result was 0.58. The difference between the gold standard and end-to-end runs indicates that knowledge resource or document retrieval can be improved to write a good essay.

In all settings, short essay performances are lower than those of long essays. This difference is attributed to the lack of keywords of the answer in short essay. In short essay, necessary important terms (mainly proper noun) are not given in contrast to long essay question.

4.1 Answer Examples

The system answers and gold standards for the example questions shown in Section 2 are following:

Gold Standard for Short Essay

It used roads around Cuzco and knotted ropes called quipu.

System Answer (Pick One) for Short Essay

Inca road system.

System Answer (Generative) for Short Essay

road developed system

Gold Standard for Long Essay

The Society of Jesus, which engaged in missionary work overseas, was also active in China, bringing information about China to Europe. The scientific revolution of 18th century Europe brought about the Enlightenment, especially in France, with its focus on reason and equality. Voltaire praised China for lacking doctrines which were contrary to reason. This was in response to Catholic control of France since the reign of Louis XIV, who abolished the Edict of Nantes, which granted Protestant the same rights as Catholics. Reynal praised China for not having hereditary nobility. His aim was to contrast France, with its fixed class system, to China, whose appointment of ministers under the imperial examination system ensured some degree of social mobility. Montesquieu, however, criticized China's tyrannical authoritarian system. By criticizing China's restriction of free speech through the Literary Inquisition, he meant to implicitly criticize France's system of absolute monarchy. In these ways, the Enlightenment criticized France's authoritarian religion, class system, and absolute monarchy, and created the philosophical foundation of the French Revolution which overturned the absolute monarchy.

System Answer for Long Essay

For de Tocqueville, the Revolution was the inevitable result of the radical opposition created in the 18th century between the monarchy and the men of letters of the Enlightenment. It was instead the French Revolution, by destroying the old cultural and economic restraints of patronage and corporatism (guilds), that opened French society to female participation, particularly in the literary sphere.All this is not to say that intellectual interpretations no longer exist. By the end of the

Table 3: System Settings

System Name	Extraction Source	Scoring Method	Short Essay
Wiki-ExtractionScore-PickOne	Wikipedia	Extraction Score	Cherry Picking
Wiki-WordSimScore-PickOne	Wikipedia	Word Similarity (Eq. 4)	Cherry Picking
Wiki-WordSimScore-Generative	Wikipedia	Word Similarity (Eq. 4)	Generative (AMR)
Wiki-SentSimScore-PickOne	Wikipedia	Sentence Similarity (Eq. 5)	Cherry Picking
GSN-WordSimScore-NA	Gold Standard	Word Similarity (Eq. 4)	N.A. (Long essay only)

Table 4: End-to-end Evaluation Result of Each System

System	Evaluation Method	Number of Questions	Mean	Max	Median	Min	Variance	Standard Deviation
FelisCatusZero	ROUGE-1	27	0.063	0.244	0	0	0.007	0.081
	ROUGE-2	27	0.009	0.067	0	0	0.000	0.018
Wiki-ExtractionScore-PickOne	ROUGE-1	27	0.118	0.261	0.143	0	0.009	0.093
	ROUGE-2	27	0.030	0.143	0	0	0.002	0.041
Wiki-WordSimScore-PickOne	ROUGE-1	27	0.123	0.32	0.1	0	0.008	0.088
	ROUGE-2	27	0.025	0.167	0	0	0.002	0.042
Wiki-WordSimScore-Generative	ROUGE-1	27	0.079	0.234	0.057	0	0.007	0.081
	ROUGE-2	27	0.013	0.105	0	0	0.001	0.026
Wiki-SentSimScore-PickOne	ROUGE-1	27	0.107	0.348	0.095	0	0.010	0.098
	ROUGE-2	27	0.023	0.174	0	0	0.002	0.043

Table 5: Comparison of End-to-end, Short and Long essay task ROUGE-1 and 2 Means.

System	End-to-end ROUGE-1 Mean	End-to-end ROUGE-2 Mean	Short Essay ROUGE-1 Mean	Short Essay ROUGE-2 Mean	Long Essay ROUGE-1 Mean	Long Essay ROUGE-2 Mean
FelisCatusZero	0.063	0.010	0.032	0.004	0.202	0.032
Wiki-ExtractionScore-PickOne	0.118	0.030	0.097	0.028	0.210	0.041
Wiki-WordSimScore-PickOne	0.123	0.023	0.105	0.021	0.203	0.040
Wiki-WordSimScore-Generative	0.079	0.025	0.051	0.007	0.203	0.040
Wiki-SentSimScore-PickOne	0.107	0.012	0.086	0.017	0.201	0.05
GSN-WordSimScore-NA					0.584	0.359

18th century, prominent French philosophers and literary personalities of the day, including Anne-Robert-Jacques Turgot, were making persuasive arguments to promote religious tolerance. The edict paved the way for the most far-reaching reforms in terms of their social consequences, including the creation of a national education system and the abolition of the imperial examinations in 1905.

5 CONCLUSIONS

In this paper, the Wikipedia based essay question answering system for world history subject question of university entrance examination was discussed. Six modules; question analysis, document retrieval, sentence extraction, sentence scoring, short essay generation, and sentence ordering are described and tested. The proposed system extracts keywords from the question text, and weights of the keywords are determined based on tf-idf score of the entire Wikipedia. Related articles are retrieved in whole Wikipedia and important sentences are extracted based on the weighted keywords. Cherry picking or generative method are attempted to generate for short essay. For a long essay, sentence ordering is used. The results of the end-to-end evaluation indicated that the proposed system succeeded to generate better essays compared with the the only reference system which uses machine translated textbooks as the knowledge resource. However, the performance difference was not statistically significant because the number of provided dataset was small. In addition, even though it should be noted that the results of the proposed system and the previous research cannot be simply compared because of the different questions, the ROUGE-1 mean score of the answers generated by the proposed system is about three times larger than that of the previous study that also uses Wikipedia, 0.0326 [7]. Failure analysis of the proposed system is future work.

ACKNOWLEDGMENTS

We thank Prof. Eric Nyberg for assistance with the development of the essay QA system.

REFERENCES

[1] 2017. *The 13th NTCIR.* http://research.nii.ac.jp/ntcir/ntcir-13/.

[2] 2017. *NTCIR-13 QA Lab-3.* http://research.nii.ac.jp/qalab/.

[3] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. 2013. API design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238* (2013).

[4] James P Callan. 1994. Passage-level evidence in document retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval.* Springer-Verlag New York, Inc., 302–310.

[5] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 335–336.

[6] Van Dang and W Bruce Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval.* ACM, 65–74.

[7] Min-Yuh Day, Cheng-Chia Tsai, Wei-Chun Chuang, Jin-Kun Lin, Hsiu-Yuan Chang, Tzu-Jui Sun, Yuan-Jie Tsai, Yi-Heng Chiang, Cheng-Zhi Han, Wei-Ming Chen, Yun-Da Tsai, Yi-Jing Lin, Yue-Da Lin, Yu-Ming Guo, Ching-Yuan Chien, and Cheng-Hung Lee. 2016. IMTKU Question Answering System for World History Exams at NTCIR-12 QA Lab2. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016.*

[8] Dheeru Dua, Bhawna Juneja, Sanchit Agarwal, Kotaro Sakamoto, Di Wang, and Teruko Mitamura. 2016. CMUQA: Multiple-Choice Question Answering at NTCIR-12 QA Lab-2 Task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016.*

[9] Sakamoto Kotaro. 2017. *FelisCatus Zero-multilingual.* https://github.com/ktr-skmt/FelisCatusZero-multilingual.

[10] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, Vol. 8. Barcelona, Spain.

[11] Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman M. Sadeh, and Noah A. Smith. 2015. Toward Abstractive Summarization Using Semantic Representations. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar (Eds.). The Association for Computational Linguistics, 1077–1086. http://aclweb.org/anthology/N/N15/N15-1114.pdf

[12] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *AAAI*. 2786–2792.

[13] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *EMNLP*, Vol. 14. 1532–1543.

[14] Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 275–281.

[15] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. *CoRR* abs/1509.00685 (2015). http://arxiv.org/abs/1509.00685

[16] Kotaro Sakamoto, Takaaki Matsumoto, Madoka Ishioroshi, Hideyuki Shibuki, Tatsunori Mori, Noriko Kando, and Teruko Mitamura. 2017. FelisCatusZero: A world history essay question answering for the University of TokyoâĂŹs entrance exam. In *Proceedings of Open Knowledge Base and Question Answering Workshop at SIGIR.* (to appear).

[17] Hideyuki Shibuki, Kotaro Sakamoto, Madoka Ishioroshi, Akira Fujita, Yoshinobu Kano, Teruko Mitamura, Tatsunori Mori, and Noriko Kando. 2016. Overview of the NTCIR-12 QA Lab-2 Task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016.*

[18] Hideyuki Shibuki, Kotaro Sakamoto, Yoshinobu Kano, Teruko Mitamura, Madoka Ishioroshi, Kelly Y. Itakura, Di Wang, Tatsunori Mori, and Noriko Kando. 2014. Overview of the NTCIR-11 QA-Lab Task. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-11, National Center of Sciences, Tokyo, Japan, December 9-12, 2014.*

[19] Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. 2005. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, Vol. 2. Amherst, MA, USA, 2–6.

[20] Di Wang, Leonid Boytsov, Jun Araki, Alkesh Patel, Jeff Gee, Zhengzhong Liu, Eric Nyberg, and Teruko Mitamura. 2014. CMU Multiple-choice Question Answering System at NTCIR-11 QA-Lab. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-11, National Center of Sciences, Tokyo, Japan, December 9-12, 2014.* http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/QALab/05-NTCIR11-QALAB-WangD.pdf

[21] Renxian Zhang. 2011. Sentence ordering driven by local and global coherence for summary generation. In *Proceedings of the ACL 2011 Student Session.* Association for Computational Linguistics, 6–11.

Applying Linked Open Data to Machine Translation for Cross-lingual Question Answering

Takaaki Matsumoto
Carnegie Mellon University
5404 GHC 5000 Forbes Ave.
Pittsburgh, Pennsylvania, USA 15213
SOC Corporation
3-16-17, Takanawa, Minato-ku
Tokyo, Japan 108-0074
takaaki.m@gmail.com

Teruko Mitamura
Carnegie Mellon University
6711 GHC 5000 Forbes Ave.
Pittsburgh, Pennsylvania, USA 15213
teruko@cs.cmu.edu

ABSTRACT

This paper addresses the methodology and its evaluation for answering cross-lingual essay questions by utilizing linked open data which assists machine translation. The question answering (QA) system studied in this paper generates English essays for the world history subject of the entrance examination of University of Tokyo. Most answers can be found in the Japanese world history textbooks. However, equivalent content of high quality English translation of the Japanese world history textbooks are not available. Therefore, we try to translate those textbooks utilizing linked open data, and to use source language knowledge resource of which content is not equivalent with the target knowledge resource.

The evaluation result indicates that the proposed method shows better performance compared with the baseline method [10] and the previous research [4]. The result of the proposed system is almost equivalent to the well designed Wikipedia based system [8].

The result of this paper concludes that 1) simple neural translation of knowledge resource does not work for domain specific cross-lingual question answering, 2) linked open data is effective to find correct translation for difficult terms in machine translation process, and 3) adding source language open knowledge resource would help even if its content is not equivalent with the target knowledge resources.

KEYWORDS

Question answering, linked open data, NTCIR-13, Wikidata

1 INTRODUCTION

Question Answering (QA) research has been done for a long time, and their successes are widely found in factoid and multiple-choice questions. However, essay question answering, which is often found in a real-world situation, is considered to be one of the most difficult QA tasks, because it is often related to a multi-document summarization task.

It is essential to have knowledge resources to solve essay QA tasks. Some domains, for example law, patent, business, and so on are highly dependent on a language or a culture, and effective knowledge resources disproportionately exist from language to language. For example, answering English essay question about Japanese business custom is not an easy task. There are three ways to solve this kind of cross-lingual QA; 1) applying machine translation to question and answer, and solving the QA task in the target language, 2) translating the target knowledge resources into the source language by machine, and solving the QA task in the source language, and 3) solving the QA in the source language using a large scale open-domain knowledge resource of the source language, hence it is a mono-lingual QA. The first option is the simplest way. However, two times machine translations, source question to target question and source answer to target answer, may reduce the translation accuracy. The second option can be a useful approach if the knowledge resources are not very large. The third option does not contain machine translation. However, since a large scale open-domain knowledge resource like Wikipedia is high signal to noise ratio, retrieving correct answer is difficult. This paper employs the second option, because the knowledge resource size of the target task is small enough.

The NTCIR-13 QA Lab is a challenge to solve the Japanese university entrance examinations (on world history) in English [3][14][13] . In the QA Lab, there are three types of questions; multiple-choice, term (factoid), and essay question. The essay questions of QA Lab are selected from the past world history examinations of University of Tokyo, Japan. University of Tokyo entrance examination is considered to be one of the most difficult examinations in Japan, and generally questions are based on the Japanese high school textbooks.

In the task, there are two types of essays; 1) short/simple essay and 2) complex/long essay. A short/simple essay question expects a short answer, which is usually a single sentence (15-60 words). Many of these questions may contain a factoid question as part of the answer. A complex/long essay question requires a longer answer, which consists of multiple sentences (225-270 words). It usually contains a longer introductory paragraph and it also contains a list of 4-9 keywords that are required to be used in the essay.

In this paper, we focus on the essay question answering for world history subject in the NTCIR-13 QA Lab-3 in English. We describe the previous challenges and performance difference between closed and open knowledge bases (Section 2), the methodology to utilize linked open data for the task in English (Section 3), results and discussions of the proposed method (Section 4), and conclude the paper (Section 5). In the Section 4, the evaluation result of the proposed system is compared with not only the baseline but also an another QA system that uses a large scale open domain knowledge base, which is mentioned as the third option in above.

2 PREVIOUS RESEARCH AND BASELINE

In the NTCIR-12 QA Lab-2 (2016) [1], Phase-1, both English and Japanese essay tasks were evaluated. The best ROUGE-1 [7] scores were quite different; the best Japanese system had approx. 0.3 [13][11], while the English system had 0.0326 [4]. This was only 1/10 of that of Japanese. One of the reasons for low scores in English can be a language barrier, because the entrance examination is based on the Japanese world history high school textbooks and no English version of them were available.

For the baseline system of this study, we use a multilingual essay question answering system developed by Sakamoto et al. [10][12]. In the baseline system, the knowledge resources they used are machine translated texts of five Japanese world history textbooks and one Japanese world history glossary published from Tokyo Shoseki and Yamakawa. The translation was attempted in 2015 with Google translate, in which the statistical translation technique was used.

3 PROPOSED METHOD

As described above, one of the most different things between Japanese and English tasks in NTCIR QA Lab was the availability of the knowledge resources. Japanese teams could use five Japanese high school textbooks, while English teams mainly used Wikipedia. In this section, we propose an essay generating system for cross-lingual question answering task that utilizes linked open data for machine translation of the knowledge resource.

3.1 Improving of Machine Translation of Native Textbooks using Linked Open Data

The proposed method attempts to improve machine translation quality of Japanese textbooks. We use a linked open data to find correct translation.

A preliminary study of Japanese exams indicated that the Japanese textbooks cover more than 80% of the questions of University of Tokyo entrance examinations. However, machine translated textbooks by Google Translate in 2015 lack many important terms and produce errors. For example, ササン朝 (Sasanian Empire) was translated as "sasan morning," because the Japanese character 朝 means both "dynasty" and "morning," and generally uses as "morning." The latest neural translation technology might be able to improve translation quality. However, we found that some nouns are mistranslated in the neural translation as follows (Table 1). Table 1 clearly shows some nouns (especially, compound noun) were mistranslated by the latest neural transition, and Wikidata.org translated them perfectly. Therefore, in order to translate difficult but important terms, we created a bilingual world history term corpus by utilizing linked open data (LOD).

3.1.1 Bilingual World History Term Corpus. In order to find the correct English translation in the Wikidata.org and build a bilingual world history term corpus, two strategies were adopted; 1) exact match or only one, 2) longer match.

The objective of the first strategy is to generate the bilingual corpus with very high precision and adequate recall. A candidate Japanese term found in the Japanese world history glossary is firstly tried exact match in Wikidata.org. If it matches, the translation word is retrieved. If it does not match exactly, then the word is searched, and if the number of search results is only one, the translation word is retrieved. If the number of the search result are greater than two, the translated results are ambiguous and they are not utilized.

The second strategy is to avoid mistranslation. This strategy would help to retrieve compound nouns correctly. Assume that the following Japanese passage in the glossary:

またキリスト教綱要によれば

(Also according to the Institutes of the Christian Religion).

Firstly, morphological analysis (MeCab [6]) is applied and tokenized text is obtained.

また|キリスト|教|綱要|に|よれ|ば

(CONJ | NP | suffix | N | case marker | V | CONJ particle).

Then, the linked open data assists translation. Translation starts with a noun or proper noun, and ends if the next word is neither a noun nor some exceptions (suffix or some symbols). At first, また, which means "also," is neither a proper noun or a noun, and therefore また is ignored. キリスト is a proper noun and the translation starts. The Wikidata.org has an exact match result of "Christ." The next word 教 is a suffix and the translation continues. キリスト教 is also found in Wikidata.org and the translation of "Christianity" is retrieved. 綱要 is also a noun and キリスト教綱要 is found in Wikidata.org and its translation of "Institutes of the Christian Religion" is saved. The next word に is a case marker, so the translation process stops. Finally, the longest translation "Institutes of the Christian Religion" word is retrieved correctly as the translation of キリスト教綱要.

By using this technique, the bilingual world history translation corpus was generated. Since the results were large, we could not examine all the results. However, we sampled the results and found that most long terms are correct and some short terms were wrong. We checked all terms of which length is less than 4 characters, and found only approx. 100 mistranslations in the results. Finally, 6,962 Japanese terms and their English translations were retrieved. In addition, approx. 2,000 English words were added from the world history ontology [5].

3.1.2 Translating Japanese World History Textbooks. The Japanese textbooks are translated in two steps; firstly by the bilingual world history term corpus described 3.1.1 and secondly by commercial translation API (Microsoft Bing Translator). At first, all terms that match with the bilingual corpus in the whole Japanese text are replaced into English terms and then a Japanese-English mixed text is generated. After that, it is translated by commercial neural translation API. In this paper, we used Microsoft Bing Translator since it translated some world history related nouns better than Google Translate as shown in Table 1. For example, a Japanese passage:

" またキリスト教綱要によれば "

is firstly translated into Japanese-English mixed text:

" また Institute of the Christian Religion によれば. "

Then, the text is translated by Microsoft Bing Translator into:

" Also according to the Institute of the Christian region. "

This is a better translation than Google Translate, " According to Christianity requirements. "An example of this process is shown in the appendix.

Table 1: Translation Examples

Japanese Term	Google Translate (2017)	Bing Translator	Wikidata	Correct Translation
林則徐	Hayashi Noriro	the zexu	Lin Zexu	Lin Zexu
欽差大臣	Minister of Ginza	Minister of the Qin	Imperial Commissioner	Imperial Commissioner
キリスト教綱要	Christianity requirements	Christian elements	Institute of the Christian Religion	Institute of the Christian Religion

3.1.3 Discussion. The proposed method has two strategies, 1) exact match or only one, and 2) longer match, to build the bilingual world history term corpus. They might be seem not to be effective to solve critical issues that may arise in the translation process, because the "exact match or only one" strategy can be regarded as avoiding of the ambiguity problem. However, based on our observations and assumptions of the translation problems of the world history textbooks, we think that the proposed strategies are effective even.

Firstly, we found that most of the mistranslating terms in the Japanese world history textbooks are very difficult and rare nouns. They are the names of a person, country, dynasty, war, treaty, and so on. Those terms are often found unambiguous ways. Some wars or treaties have alias names. However, since we can write down only one name in the answer in general and alias name is not often asked, translation to the alias name is not necessary.

Secondly, the combination of the "exact match or only one" and the second strategy of the "longer match" often helps to solve ambiguity problems. Let's look at the example of オスマン帝国は (in English, Ottoman empire is). By the morphological analysis of the MeCab, we obtain a chain of morphemes of オスマン/帝国/は (NP/N/Particle). The system tries exact match of the first word オスマン in Wikidata.org. However, it is ambiguous and has no exact match. Then, because of no exact match, searching in Wikidata.org is attempted. We have many search results, Ottoman Empire, Osman I, Ottoman Dynasty, Ottoman Turkish, and so on. These translations can be correct if only the word of オスマン is given. This kind of ambiguity can be solved by contexts. However, we have the another noun of 帝国, which succeeds to the オスマン. The compound noun of オスマン帝国 gets the exact match of the Ottoman Empire. We still have many search results for オスマン帝国, if searching in Wikidata.org is attempted. However, exact match has precedence over searching in our algorithm, and the ambiguity problem does not happen if the exact match is succeeded.

Searching in Wikidata.org makes sense when the term has alias names, including orthographic variants. As we pointed before, we have some aliases for word history terms. Especially, Japanese has Romanization and it often generates many similar aliases. For example, “Sasanian Empire” is represented as ササン朝 in the textbooks we used, but, the de-facto translation is considered to be サーサーン朝, which uses to macrons (there are many orthographic variants for foreign originated terms in Japanese Katakana). Hence, ササン朝 fails exact match in Wikidata.org because it only checks the title of the article. However, the articles in Wikidata.org contains alias field and we can find “Sasanian Empire” when we use the search of ササン朝. Another example for this problem is じゃがいも飢饉 (Great Irish Famine). Since じゃがいも飢饉 is a common noun compound, Google translate mistranslates “Potato famine,” which is translations of じゃがいも and 飢饉. However, Wikidate.org can find correct translation for not only the de-facto term of じゃがいも飢饉 but also its alias name of アイルランド大飢饉 (Great Irish Famine). We can say the proposed strategies can handle the translation problem of the orthographic variants or alias names of the source language (Japanese) correctly.

Table 2: Comparison between LOD assisted Machine Translation and Simple Machine Translation

	Number of words translated by LOD	LOD Failure and Bing Success	Bing Failure and LOD Success
Sample 1	33	1	2
Sample 2	40	3	10
Sample 3	22	1	3
Sample 4	21	1	9
Sample 5	42	3	7

Another discussion for the proposed method can be words that are not in the Wikidata.org are not usable (as mentioned in 3.1.1). We used the language link data of the Wikidata.org which is equivalent with the inter-language link of the Wikipedia articles to find correct translation. Some articles of the Wikipedia are deep-rooted in the culture and tradition and few language links can be found, and some words are clearly not in Wikipedia. However, since the question answering task in this paper deals with the world history subject of a university entrance examination, we think that the coverage of the Wikidata.org is considered to be enough.

We analyzed 5 sample articles of a textbook, which becomes approx. 250 words in English after translation (the original articles have about 500 characters in Japanese). We counted the number of words translated by the bilingual world history term corpus (LOD assisted machine translation), and checked their translation quality. Table 2 shows the result. In all five sampled articles, approximately from 20 to 40 words of each article were translated from Japanese to English using the bilingual world history term corpus. A few (from 1 to 3) words of each article were found to be mistranslated. About the half of them could be translated correctly if the Bing Translator is used directly, but the another words cannot be translated by both of the corpus (Wikidata) and Bing Translator. When we directly applied Bing Translator to the sample articles, we had many mistranslations for the words that were translated by the bilingual corpus correctly. This result indicates that the pre-translation by the proposed bilingual world history term corpus is

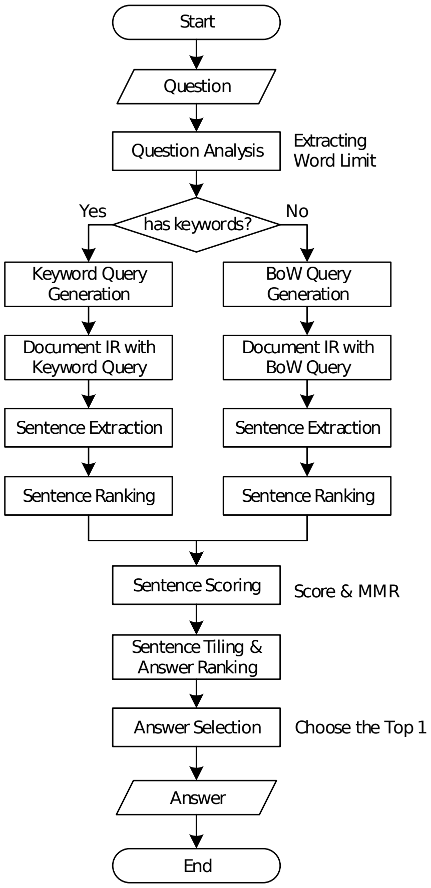


Figure 1: System Flowchart.

very effective for the machine translation of the textbooks to translate rare nouns correctly. On the other hand, we found some effects of the pre-translation process. Some sentences can lose coherency, and the translation quality of some words improves or worsens. These analyses are future research.

3.2 Additional Domain Specific Open Knowledge

Since the translation of Japanese textbooks is done by machine translation, mistranslations are inevitable. Therefore, we add one public English world history textbook from Boundless.com [2]. While some public English world history textbooks are available in PDF format in on-line, the textbook of Boundless.com is a HTML based and easy to use for natural language processing task.

3.3 System Description

Fig. 1 shows the system flowchart of this method. The system flow is following.

(1) At first, the question data is given in XML format.
(2) The question data is analyzed by the question analysis module, and the maximum answer length is obtained.
(3) The system has a different IR strategies for question type. If the question has keywords that are required to be used in the essay, the question is a complex/long essay. Otherwise, the question is regarded as a short/simple essay.
(4) Query data for IR is generated. For long essays, the keywords in the question are used. For short essays, the bag of word (BoW) of the question sentences are adopted.
(5) Using the query, documents (set of passages) are retrieved from the knowledge resources.
(6) Sentences are ranked by the IR scores.
(7) Sentences scoring module gives a score which indicates the relevance or entailment for the question to the extracted sentences.
(8) Scored tiling module generates essays by changing order of the extracted sentences. The score of an essay candidate is summation of the sentence scores in the essay.
(9) The top 1 score essay is chosen as the answer.
(10) The answer XML data is generated.

The baseline system uses following scoring method by default:

$$\text{Score} = \frac{k_m}{m} \tag{1}$$

where k_m is the number of keywords in the sentence, and m is the number of words of the sentence. All keywords and words of the sentence are stemmed. Stop words and punctuations are removed before calculation.

Eq.1 measures the density of the keywords in a sentence. However, not always the given keywords and words in the sentence match exactly. Some words of the answer sentence could be similar to the given keywords. Hence, word level similarity between retrieved or given keywords and an extracted sentence is calculated as follows:

$$\text{Score} = \sum_{i=1}^{m} \frac{\max(w_i \cdot k_1, w_i \cdot k_2, ...w_i \cdot k_n)}{\log m} \tag{2}$$

where, m is the number of words in the sentence except stop words and punctuations, n is the number of keywords, w_i is the $i-$th word vector of the sentence, and k_j is the $j-$th keyword vector. Word embedding is given by GloVe [9]. Using the score, answer candidates are generated and their scores are also given by just summation of the sentence score. Finally, the top 1 essay is selected as an answer and answer XML file is outputted.

4 RESULT AND DISCUSSION

The proposed methods are evaluated using the NTCIR-13 QA Lab-3 official phase-1 dataset, which contains 5 long/complex and 22 short/simple essay questions and ground truths [3]. In the QA Lab, evaluation is done by human experts, ROUGE method and Pyramid method [3][13]. In this paper, ROUGE-1 and 2, unigrams and bigrams to compare the essay to a set of gold-standard essays, are used for evaluation. Sample questions, gold standards and system answers are shown in the appendix.

Table 3 shows the ROUGE-1 and ROUGE-2 evaluation results of the baseline and prosed method for the dataset.

The system A shows the combination of the baseline system and the baseline knowledge resources (machine translated Japanese textbooks using Google Translate in 2015). The system B shows the combination of the baseline system and the neural machine

Table 3: End-to-end Evaluation Result of Each System

System	Evaluation Method	Number of Questions	Mean	Max	Median	Min	Variance	Standard Deviation
(A) Baseline	ROUGE-1	27	0.063	0.244	0	0	0.007	0.081
	ROUGE-2	27	0.009	0.067	0	0	0.000	0.018
(B) Baseline	ROUGE-1	27	0.056	0.260	0	0	0.010	0.077
+NMT	ROUGE-2	27	0.004	0.054	0	0	0.000	0.011
(C) Baseline	ROUGE-1	27	0.081	0.375	0.054	0	0.010	0.100
+LNMT	ROUGE-2	27	0.011	0.064	0	0	0.000	0.021
(D) Baseline	ROUGE-1	27	0.076	0.225	0.063	0	0.010	0.075
+LNMT + WS	ROUGE-2	27	0.012	0.118	0	0	0.001	0.027
(E) Baseline	ROUGE-1	27	0.128	0.485	0.105	0	0.015	0.122
+LNMT +WS +ET	ROUGE-2	27	0.028	0.176	0	0	0.003	0.050
(F) Wikipedia-based	ROUGE-1	27	0.123	0.320	0.1	0	0.008	0.088
	ROUGE-2	27	0.025	0.167	0	0	0.002	0.042

translated (NMT) textbooks, and it is worse than that of the baseline system and baseline knowledge resource. One of the reasons of the difference is the mistranslation of some rare terms, as pointed out in the section 3. The system C shows the combination of the baseline system and linked open data (Wikidata) which assisted neural machine translated textbook (LNMT). When LOD assisted neural machine translated textbooks are used, the score was improved. Since the ROUGE-1 is based on unigrams to compare to the gold-standard, correct words in an answer existed is very important. In addition, the LOD assisted translation can give correct English entity names. Therefore, system C improved the ROUGE score effectively.

The system D and E adopt word similarity based sentence scoring (WS). The system D gets almost the same ROUGE-1 and ROUGE-2 means compared with those of the system C. However, when the English textbook (ET) is added to the knowledge resource (system E), it has the best ROUGE-1 and ROUGE-2 means. It should be noted that the number of the question is only 27 (5 long essays and 22 short essays). Since the NTCIR QA Lab uses the real past entrance examination of University of Tokyo, the provided data was very small. The performance differences are not statistically significant when the standard deviations are considered.

System F is the reference system developed for the same task (NTCIR-13 QA Lab-3) [8] which uses whole English Wikipedia as the knowledge resource. It employs carefully designed keyword weighting for document retrieval and sentence extraction to overcome the high signal to noise ratio of the whole Wikipedia. The proposed system in this paper has almost equal ROUGE-1 and 2 means to the system F. In addition, even though it should be noted that the results of the proposed system and the previous research cannot be simply compared because of the different questions, the best ROUGE-1 mean of the proposed system is about four times larger than that of the previous study that also uses Wikipedia (0.0326 in ROUGE-1 mean) [4].

In summary, the reasons for the better ROUGE-1 and ROUGE-2 means of the proposed method compared with that of the baseline are attributed to the accurate named entities of the knowledge resources and the similarity measurement in the sentence scoring process.

Table 4: Short and Long Essays

System	All Essay ROUGE-1 Mean	Short Essay ROUGE-1 Mean	Long Essay ROUGE-1 Mean
(A) Baseline	0.063	0.032	0.202
(E) Baseline +LNMT +WS + ET	0.128	0.114	0.190

4.1 Comparison of the Short and Long Essays

Table 4 shows the comparison of the short and long essay ROUGE-1 means. It clearly indicates that the performance improvement of the proposed methods, compared with the baseline comes from short essay.

The short essay ROUGE-1 means of both proposed methods are almost half or less than those of long essays. One of the reasons of this gap between short and long essay ROUGE mean can be attributed to the short essay question answering scheme. As described in the section 1, the answer of the short essay question often contains factoid answers as a part of the essay (i.e. “ In 30 English words or less, indicate the name of this Merovingian dynasty king and explain what kind of religion he converted to.”). Since the QA systems studied in this paper generate essays by BoW search based on the question, the answer of the factoid part is often unsolved. In addition, from the aspect of the probability, getting ROUGE-1 score in a long essay is easier than short essay. Generally long essay contains 5-10 sentences, and if one of them matches to the part of the gold standard, the system answer can get non-zero score. However, in short essays, the answer usually have only one sentence. Therefore, long essay answer has approx. 5-10 times larger chance to get positive ROUGE score than short essay.

5 CONCLUSIONS

In this paper, the methodology and its evaluation results for essay question answering for a narrow domain by utilizing linked open data was discussed. The proposed method translates narrow domain knowledge resources (Japanese world history textbooks) by utilizing Wikidata. The evaluation result indicated that the proposed method showed better performance compared with the baseline method [10] and the previous research [4]. The result of the proposed system was almost equivalent to the well designed Wikipedia based system [8].

The result of this paper concludes that 1) simple neural translation of knowledge resource does not work for domain specific cross-lingual question answering, 2) linked open data is effective to find correct translation for difficult terms in machine translation process, and 3) adding source language open knowledge resource would help even if its content is not equivalent with the target knowledge resources.

A LINKED OPEN DATA ASSISTED MACHINE TRANSLATION EXAMPLE

At first, we extract text from Japanese world history textbooks as follows:

> イギリスで増大しつづける中国茶（紅茶）の消費に対して，イギリス東インド会社はしだいに銀による支払いが追いつかなくなっていた。そこで，１８世紀末から，イギリスはインドでアヘンの専売制を始め，専売による財源の増加とアヘンを中国に売却することによって，茶の支払いにあてようとした。１８３９年，アヘン弛禁派をおさえ，厳禁派の林則徐が欽差大臣として広州に派遣され，アヘン密輸問題の解決にあたった。彼は外国商人がもつアヘンを没収してそれを廃棄した。アヘン貿易商人はこれに強く反発し，イギリス議会ではグラッドストンらによる“恥ずべき戦争”という反対にあったが，９票差で戦争を決定した。イギリス軍は沿岸の各地で清軍をやぶりながら北上し，１８４２年清朝と南京条約を結んだ。イギリスにつづき，１８４４年にアメリカ（望厦条約）とフランス（黄埔条約）も同様な条約を結び，清朝に条約の完全履行をせまった。しかし，華夷思想（中華思想）にもとづく朝貢外交の様式と異なるこの条約外交は，広州へのイギリス人の入城に対して地方官僚がそれを拒否するなど多くの摩擦を発生させることとなった。

Then, the bilingual world history term corpus is applied to the text:

> United Kingdomで増大しつづけるChina茶（紅茶）の消費に対して，East India Companyはしだいに銀による支払いが追いつかなくなっていた。そこで，１８世紀末から，United KingdomはIndiaでアヘンの専売制を始め，専売による財源の増加とアヘンをChinaに売却することによって，茶の支払いにあてようとした。１８３９年，アヘン弛禁派をおさえ，厳禁派のLin ZexuがImperial CommissionerとしてGuangzhouに派遣され，アヘン密輸問題の解決にあたった。彼は外国商人がもつアヘンを没収してそれを廃棄した。アヘン貿易商人はこれに強く反発し，United Kingdom議会ではGladstoneらによる“恥ずべき戦争”という反対にあったが，９票差で戦争を決定した。British Armed Forcesは沿岸の各地で清軍をやぶりながら北上し，１８４２年清朝とTreaty of Nanjingを結んだ。United Kingdomにつづき，１８４４年にUnited States of America（Treaty of Wanghia）とFrance（Treaty of Whampoa）も同様な条約を結び，清朝に条約の完全履行をせまった。しかし，Sinocentrism（Sinocentrism）にもとづく朝貢外交の様式と異なるこの条約外交は，GuangzhouへのBritish peopleの入城に対して地方官僚がそれを拒否するなど多くの摩擦を発生させることとなった。

Finally, machine translation (in this paper, we used Microsoft Bing Translator) is applied:

> The East India Company was gradually unable to keep up with silver payments for the consumption of a growing *China tea (tea)* in United Kingdom. So, since the end of the eighteenth century, United Kingdom began to pay for tea by starting with the monopoly system of opium in India and by selling opium to China. In 1839, the prohibition of opium Sag, Lin Zexu was dispatched to Guangzhou as Imperial Commissioner to solve the problem of opium smuggling. He confiscated the opium of the foreign merchant and disposed of it. The opium trade merchant rallied strongly against this, and the *United Kingdom Congress* was opposed to the "shameful war" by Gladstone, but the war was decided by nine votes. British armed Forces north while veils the Qing army in various parts of the coast, and tied the *Senyi morning* and Treaty of Nanjing in 1842. United Kingdom, United States of America (Treaty of Wanghia) and France (Treaty of Whampoa) signed a similar treaty to the Qing Dynasty in 1844 to fully implement the Treaty. However, this treaty diplomacy which differs from the style of tribute diplomacy based on Sinocentrism (Sinocentrism) has caused a lot of friction that local bureaucrats refuse it for the entry of British people to Guangzhou.

In this machine translated text, there are three translation errors. “China tea,” “United Kingdom Congress” and “Senyi mornig” should be “Chinese tea” “British Parliament” and “Qing Dynasty,” respectively.

However, if we use machine translation only, there are six mistranslations; “the zexu,” “Minister of the Qin,” “Senyi morning,” “the Treaty,” “Huang Tai Treaty,” and “Chinese thought,” as follows:

> The British East India Company was gradually unable to keep up with silver payments for consumption of growing Chinese tea. So, since the end of the eighteenth century, the British tried to pay for the tea by starting the monopoly system of opium in India, increasing the financial resources and selling opium to China. In 1839, the prohibition of opium-sag, and

> *the zexu* of the Forbidden faction was dispatched to Guangzhou as *Minister of the Qin*, and the settlement of the opium smuggling problem was resolved. He confiscated the opium of the foreign merchant and disposed of it. The opium trade merchant rallied strongly against this, and the British Parliament was opposed to the ’ shameful war ’ by the Gladstone, but the war was decided by nine votes. The British Army veils the Qing army in various parts of the coast, and it tied the Nanjing Treaty with *Senyi morning* in 1842. In 1844, the United States (*the Treaty*) and France (*Huang Tai Treaty*) signed a similar treaty to the United Kingdom, and the Qing Dynasty concluded the full implementation of the Treaty. However, this treaty diplomacy, which differs from the style of tribute diplomacy based on *Chinese thought*, has caused a lot of friction, such as local bureaucrats refusing to enter the British into Guangzhou.

Compared with the linked open data assisted translated text, the mistranslations in this text are serious. For example, the name of treaty or person name are vanished or wrong. Since the names of treaty, person, dynasty, and so on often appear as the required keywords in answer or the important keywords for document retrieval in the question, losing this kind of terms can cause a serious problem.

ACKNOWLEDGMENTS

The authors thank Mr. Kotaro Sakamoto and Prof. Eric Nyberg for assistance with the development of the essay QA software.

REFERENCES

[1] 2016. *NTCIR-12 QA Lab-2.* http://research.nii.ac.jp/qalab/qalab2/index.html.

[2] 2017. *Boundless.com.* https://www.boundless.com/.

[3] 2017. *NTCIR-13 QA Lab-3.* http://research.nii.ac.jp/qalab/.

[4] Min-Yuh Day, Cheng-Chia Tsai, Wei-Chun Chuang, Jin-Kun Lin, Hsiu-Yuan Chang, Tzu-Jui Sun, Yuan-Jie Tsai, Yi-Heng Chiang, Cheng-Zhi Han, Wei-Ming Chen, Yun-Da Tsai, Yi-Jing Lin, Yue-Da Lin, Yu-Ming Guo, Ching-Yuan Chien, and Cheng-Hung Lee. 2016. IMTKU Question Answering System for World History Exams at NTCIR-12 QA Lab2. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016.*

[5] Ai Kawazoe, Yusuke Miyao, Takuya Matsuzaki, Hikaru Yokono, and Noriko Arai. 2013. World history ontology for reasoning truth/falsehood of sentences: Event classification to fill in the gaps between knowledge resources and natural language texts. In *JSAI International Symposium on Artificial Intelligence.* Springer, 42–50.

[6] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis.. In *EMNLP*, Vol. 4. 230–237.

[7] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, Vol. 8. Barcelona, Spain.

[8] Takaaki Matsumoto, Francesco Ciannella, Fadi Botros, Evan Chan, Cheng-Ta Chung, Keyang Xu, Tian Tian, and Teruko Mitamura. 2017. Wikipedia Based Essay Question Answering System for University Entrance Examination. In *Proceedings of Open Knowledge Base and Question Answering Workshop at SIGIR.* (to appear).

[9] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *EMNLP*, Vol. 14. 1532–1543.

[10] Kotaro Sakamoto. 2017. *FelisCatus Zero-multilingual.* https://github.com/ktr-skmt/FelisCatusZero-multilingual.

[11] Kotaro Sakamoto, Madoka Ishioroshi, Hyogo Matsui, Takahisa Jin, Fuyuki Wada, Shu Nakayama, Hideyuki Shibuki, Tatsunori Mori, and Noriko Kando. 2016. Forst: Question Answering System for Second-stage Examinations at NTCIR-12 QA Lab-2 Task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016.*

[12] Kotaro Sakamoto, Takaaki Matsumoto, Madoka Ishioroshi, Hideyuki Shibuki, Tatsunori Mori, Noriko Kando, and Teruko Mitamura. 2017. FelisCatusZero: A world history essay question answering for the University of Tokyo ’ s entrance exam. In *Proceedings of Open Knowledge Base and Question Answering Workshop at SIGIR.* (to appear).

[13] Hideyuki Shibuki, Kotaro Sakamoto, Madoka Ishioroshi, Akira Fujita, Yoshinobu Kano, Teruko Mitamura, Tatsunori Mori, and Noriko Kando. 2016. Overview of the NTCIR-12 QA Lab-2 Task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016.*

[14] Hideyuki Shibuki, Kotaro Sakamoto, Yoshinobu Kano, Teruko Mitamura, Madoka Ishioroshi, Kelly Y. Itakura, Di Wang, Tatsunori Mori, and Noriko Kando. 2014. Overview of the NTCIR-11 QA-Lab Task. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-11, National Center of Sciences, Tokyo, Japan, December 9-12, 2014.*

Chronological and Geographical Measures for Evaluation of World History Essay QA in University Entrance Exams

Hideyuki Shibuki
Yokohama National University
shib@forest.eis.ynu.ac.jp

Kotaro Sakamoto
Yokohama National University
National Institute of Informatics
sakamoto@forest.eis.ynu.ac.jp

Madoka Ishioroshi
National Institute of Informatics
ishioroshi@nii.ac.jp

Akira Fujita
Yokohama National University
fujita@ynu.ac.jp

Yoshinobu Kano
Shizuoka University
kano@inf.shizuoka.ac.jp

Teruko Mitamura
Carnegie Mellon University
teruko@cs.cmu.edu

Tatsunori Mori
Yokohama National University
mori@forest.eis.ynu.ac.jp

Noriko Kando
National Institute of Informatics
The Graduate University for
Advanced Studies (SOKENDAI)
kando@nii.ac.jp

ABSTRACT

We propose a method for measuring chronological and geographical consistency of the world history essays in Japanese university entrance exams. On observing several model answer essays, we found that an essay's uniformity, ordering and cooperability were important features of a well-formed paper, and we introduced them into our method. The experimental result shows a weak positive correlation between the scores measured by the proposed method and the scores estimated by a human expert in world history.

KEYWORDS

essay QA, automated evaluation, chronological and geographical measures, world history, university entrance exams

1 INTRODUCTION

Research on real-world complex question-answering (QA) has flourished in recent years [1]. In the QA Lab tasks [11, 12] at the NTCIR workshop,[1] the current problems and solutions in QA technologies have been investigated using the world history questions in Japanese university entrance exams and their English translation. Japanese university entrance exams include various types of questions such as multiple-choice, fill-in-the-blank, true-or-false, map understanding, chronological reordering, short-answer, and essay questions. Above all, essay QA is the most challenging, and still has many open problems, such as the evaluation of essays that QA systems generated. Although there is a way of evaluation by human experts in world history, it takes considerable time and cost. In the case of the QA Lab, evaluation of 46 essays by an expert who teaches world history took around a month and about 500,000 yen (4,500 USD). Therefore, a new method is required.

Because essay generation is regarded as a kind of query-biased summarization, the measures for evaluating summaries using gold-standard data can be applied to essay evaluation. In the QA Lab, the ROUGE family [6] and the Pyramid method [8, 10] are used for grading essays besides a human expert's evaluation. A positive correlation between these grades and those provided by humans was between moderate and weak, and the ranking order by the measures was not always concordant with the ranking order given by the human marks. Therefore, we investigated more appropriate measures for evaluating world history essays in Japanese university entrance exams.

For evaluating summaries, the linguistic well-formedness and the relative responsiveness were used in the DUC workshops.[2] The content, readability/fluency, and the overall responsiveness were used at the Guided Summarization tasks[3] in the TAC workshops. These measures are important for evaluating world history essays in university entrance exams. However, the linguistic well-formedness and readability/fluency were scored arbitrarily by human assessors, while the content was methodologically scored by the ROUGE family and the Pyramid method, among others. We would like to methodologically give other scores based on merits other than the content. For evaluating world history essays, chronological and geographical consistency is important as a kind of semantic consistency. However, how to evaluate these is not obvious. What measures should be taken for chronological or geographical consistency? How should the chronological measures and the geographical measures be harmonized? In this paper, we propose a method for measuring chronological and geographical consistency of world history essays, and examined the method using essays submitted to the QA Lab.

The main contributions of this paper are as follows: (i) to clarify the features of well-formed world history essays in terms of the chronological information and the geographical information, (ii) to introduce a new scoring method based on the features to evaluate the well-formedness of world history essays.

The rest of this paper is organized as follows. Section 2 describes the features of essay questions for world history in Japanese university entrance exams. Section 3 describes the features of model

[1] http://research.nii.ac.jp/ntcir/index-en.html

[2] http://duc.nist.gov/duc2007/tasks.html

[3] http://www.nist.gov/tac/2011/Summarization/Guided-Summ.2011.guidelines.html

answer essays and the hypotheses about what constitutes a well-formed essay. Section 4 describes a method based on the hypotheses. Section 5 describes the experimental results and give them consideration. Section 6 briefly overviews related work, and describes the utility of our method. Section 7 is the conclusion.

2 ESSAY QUESTION OF WORLD HISTORY

Figure 1 shows an example of an essay question for world history, which is an English translation from the original Japanese version. The question contains additional text besides the main essay topic: "How did political authorities around the world handle religion, religious schools, and people affiliated with them within their territories?" The first paragraph gives background information, and the texts below the essay topic are the constraints for writing the essay. The constraints include a length limitation of "no more than 20 lines," a geographical condition of "West Europe, West Asia and East Asia," a chronological condition of "up to and including the first half of the 18th century," the keywords that must be used in the essay, and other associated conditions. The chronological condition and the geographical condition prove the importance of chronological and geographical consistency.

Note that we distinguish essay questions from short-answer questions in terms of description length. The length of essay is more than ten lines, while the length of short answer is a few lines. Not many universities give essay questions, and the number of essay questions in an exam is usually one or two. This means that it is impossible for a statistical approach to prepare enough training data.

3 WELL-FORMED WORLD HISTORY ESSAY

3.1 Structure

In general, a world history essay is a sequential description of historical events (HEs). A HE has both chronological information and geographical information. Let us consider how this is written. While the chronological information can be easily put in a linear order from the past to the future, the geographical information is not easy to be determinately put in a linear order because of the spatial extent. Based on the study of several model answer essays from past university entrance exam collections, the general structure of the essays follows one of two approaches: (a) disregarding geographical information, all HEs are described in chronological order, and (b) grouping HEs by the geographical information. In both, information is described in chronological order. If the former is regarded to be grouped by geographical information from "the whole world," there is no difference between the two manners; that is, both are descriptions in chronological order for HEs in a particular area. We defined a sequence of HEs with the same geographical information as a geographical section (GS). GSs could be nested hierarchically. For example, a GS of Europe may contain GSs such as England, France, and Germany, and the GS of England may contain GSs such as London, Birmingham and Manchester.

From the above, we built the following hypotheses for the structure of world history essay.

(H1) An essay is a GS.
(H2) A GS can consist of more than one sub-GSs that is in the parent GS.
(H3) HEs in a GS are put in chronological order.

3.2 Uniformity

Let us consider the uniformity of GSs in a GS. If GSs of the East Midlands, Paris and German are placed on the same level in a GS of Europe, they are incongruous even though they are all parts of Europe. This is because they are in different levels of a geographical category, such as country, region, and city. Therefore, well-formed essay require the uniformity of geographical category level. In addition, if England is described with hundreds of words while France and Germany are respectively described with a dozen words, there is incongruity even though they are in the same geographical category level. This is because their quantities of description are imbalanced. Therefore, well-formed essay seems to require the uniformity of quantity.

We built the following hypotheses for the uniformity of GSs.

(H4) GSs placed on the same level in a GS are in the same level of geographical category.
(H5) GSs placed on the same level in a GS are described in the same quantity.

3.3 Ordering

Let us consider the ordering of HEs in a GS. HEs in well-formed essays are generally described in chronological order. Note that the occurrence order of HEs does not always correspond with the descriptive order of an essay. Since the chronological information of an HE has a beginning and ending in a range, the occurrence order relation between HEs is either non-overlapping, partially overlapping or inclusive as shown in Figure 2. In all relations, the beginning of the HE e_1 precedes the beginning of the HE e_2. However, in the inclusion relation, e_1 may be described after e_2 such as "The Treaty of Nanking ended the First Opium War." Therefore, we assume that the describing order of HEs in the inclusion relation is free to the chronological order. Next, let us consider the ordering of GSs in a GS. The describing order of GSs is free relative to the chronological order. However, for example, the describing order of Athens, Rome, Cairo, Baghdad, Beijing and Shanghai seems to be better than the order of Athens, Baghdad, Beijing, Cairo, Rome and Shanghai. This is because GSs relating to each other are placed closely. We assume that the relativity is approximated by the geographical distance.

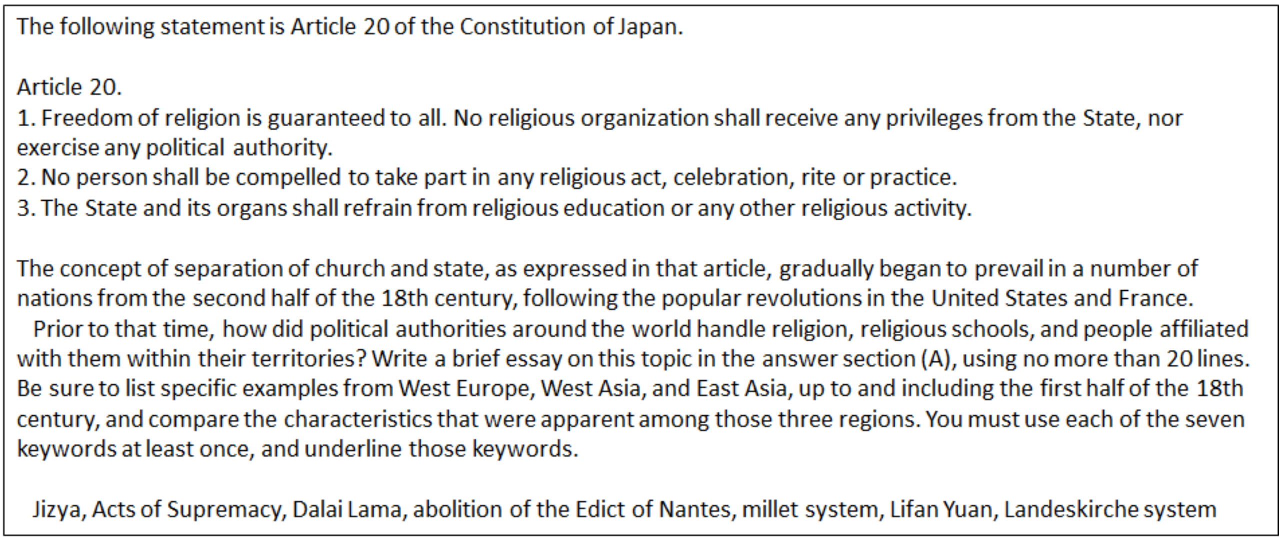
We built the following hypotheses for the ordering in a GS.

(H6) As an exception to the hypotheses (H3), an HE can be described both before and after another HE if they are in the inclusion relation.
(H7) GSs in a GS are described in the order of short geographical distance.

The hypothesis (H6) is the complement of the hypothesis (H3).

3.4 Cooperability

Let us consider the cooperability of a world history essay to question constraints in terms of the chronological and the geographical information. As described in Section 2, world history essay questions give chronological and geographical conditions such as "up to and including the first half of the 18th century" and "West Europe, West Asia and East Asia." In this case, if an essay describes only the



The following statement is Article 20 of the Constitution of Japan.

Article 20.
1. Freedom of religion is guaranteed to all. No religious organization shall receive any privileges from the State, nor exercise any political authority.
2. No person shall be compelled to take part in any religious act, celebration, rite or practice.
3. The State and its organs shall refrain from religious education or any other religious activity.

The concept of separation of church and state, as expressed in that article, gradually began to prevail in a number of nations from the second half of the 18th century, following the popular revolutions in the United States and France.

Prior to that time, how did political authorities around the world handle religion, religious schools, and people affiliated with them within their territories? Write a brief essay on this topic in the answer section (A), using no more than 20 lines. Be sure to list specific examples from West Europe, West Asia, and East Asia, up to and including the first half of the 18th century, and compare the characteristics that were apparent among those three regions. You must use each of the seven keywords at least once, and underline those keywords.

Jizya, Acts of Supremacy, Dalai Lama, abolition of the Edict of Nantes, millet system, Lifan Yuan, Landeskirche system

Figure 1: An example of essay question of world history

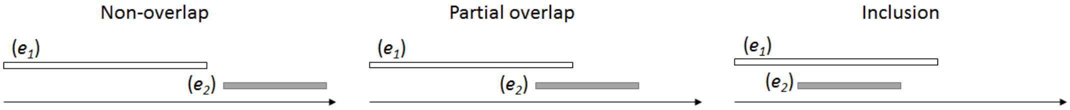


Figure 2: The pattern of chronological overlap

ancient histories of West Europe, West Asia and East Asia, the essay satisfies the conditions logically. However, it does not reflect the question intention. Since question answering is a kind of conversation, a well-formed essay will observe the cooperative principle in conversation, known as Grice's Maxims [5], which consist of quantity, quality, relation, and manner. The essay which describes only the ancient histories violates the maxim of quantity, and the cooperative essay should describe at least one HE of the 18th century. The geographical information is also similar. For example, an essay describing only "West Europe and West Asia" violates the maxim of quantity, and the cooperative essay should describe at least one HE for each area of the geographical condition. Note that a GS that is a part of an essay can violate the maxim of quantity even though the essay is cooperative. For example, the GS of West Europe in a cooperative essay may not describe all countries in West Europe. We assume that the chronological cooperability is observed in all GSs while the geographical cooperability is observed in only a GS corresponding to the essay. For a GS, we defined a period from the beginning of the earliest HE to the end of the latest one as a period of the GS. The smallest geographical range, including where the HEs in a GS occurred, was defined as the range of the GS. We assume that the observance of the maxim of quantity is approximated to the coverage of the period and the range of GSs.

We built the following hypotheses for the cooperability on the chronological and the geographical conditions in questions.

(H8) A period of a GS covers the period of the chronological condition as justly as possible.
(H9) A range of a GS corresponding to the essay covers the range of the geographical condition as justly as possible.

4 PROPOSED METHOD

4.1 Outline

In order to methodologically evaluate the well-formedness of world history essays in terms of the chronological and the geographical information, we proposed a scoring method based on the hypotheses described in Section 3. Note that the proposed method does not take into account the truth of the content. The fusion of our score and the content score measured by the ROUGE family, the Pyramid method, and others, is future work.

Figure 3 shows the outline of the proposed method. First, the input essay is segmented into HEs by punctuation marks. A HE is represented by a set of named entities extracted from the segment. Some named entities evoke the chronological and/or the geographical information. For example, "Napoleon Bonaparte" evokes the chronological information "from 15 August 1769 to 5 May 1821" and the geographical information "France." Because exam cram books cover such information, we constructed a database of world history terms based on the world history glossary published by Yamakawa Shuppan-sha.[4] Using the database, the named entities are converted into chronological and geographical information. Using both chronological and geographical information sets, the period and the range of the segment are respectively determined in the same way as that of the GS described in 3.4. They are regarded as the chronological and geographical information of the HE. Then, all hierarchical structures of GSs that can be gotten from the essay are listed. After scoring the HEs for each hierarchical structure, the maximum score is selected as the final score for the essay in order to select the most plausible hierarchical structure.

[4] http://www.yamakawa.co.jp/ (in Japanese)

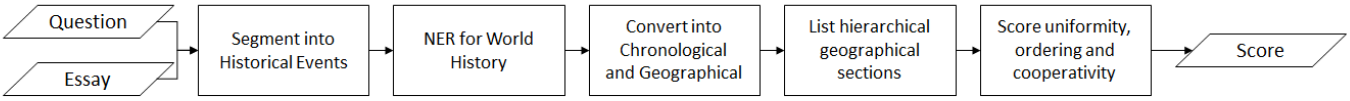

Question
Essay
Segment into Historical Events
NER for World History
Convert into Chronological and Geographical
List hierarchical geographical sections
Score uniformity, ordering and cooperativity
Score

Figure 3: The outline of the proposed method

4.2 Scoring

GSs in a hierarchical structure are classified into terminal and non-terminal sections. A terminal section means an HE sequence without hierarchical structure, and likewise a non-terminal section can be divided into several GSs. We defined a non-terminal section corresponding to the essay as the root section. A GS s is defined as a paired HE sequence $E = (e_1, e_2, \cdots, e_m)$ and GS sequence $SS = (s_1, s_2, \cdots, s_n)$. If SS is an empty tuple, then the GS is a terminal section. HEs in a sub-GS are shared with the superordinate GS, and E of non-terminal sections are not empty. For a question, the chronological condition CC is defined as a pair of the beginning time bt and the ending time et, and the geographical condition GC is defined as a geographical entities set $\{g_1, g_2, \cdots, g_k\}$.

Based on the hypothesis (H2), the score sc for a GS to a question is recursively calculated by the following expressions.

$$sc(E, SS, CC, GC) = \begin{cases} sc_T(E, CC) & \text{if it is a terminal section} \\ sc_N(E, SS, CC)sc_{GC}(E, GC) & \text{if it is the root section} \\ sc_N(E, SS, CC) & \text{otherwise} \end{cases} \tag{1}$$

$$sc_T(E, CC) = sc_{CO}(E)sc_{GO}(E)sc_{CC}(E, CC) \tag{2}$$

$$sc_N(E, SS, CC) = \frac{1}{|SS|}sc_{GU}(SS)sc_{QU}(SS) \sum_{i=1}^{|SS|} sc(events(s_i), sections(s_i), CC, GC) \tag{3}$$

where $sc_{CO}()$ and $sc_{GO}()$ are functions to score the chronological ordering and the geographical ordering described in 3.3, $sc_{CC}()$ and $sc_{GC}()$ are functions to score the chronological cooperability and the geographical cooperability described in 3.4, $sc_{GU}()$ and $sc_{QU}()$ are respectively functions to score the geographical uniformity and the quantity uniformity described in 3.2, and $events(s)$ and $sections(s)$ are functions to return an HE sequence and a GS sequence included in a GS s, respectively. We designed the scoring functions to be normalized into the range $[0, 1]$, which are described in 4.2.1 to 4.2.3.

4.2.1 *Ordering Score.*

Based on the hypothesis (H3), using the correlation between the describing order and the chronological order, the chronological ordering score sc_{CO} is calculated by the following expression.

$$sc_{CO}(E) = \frac{K - L}{K + L} \tag{4}$$

where K is the number of concordant pairs of HEs in E, and L is the number of discordant pairs. The expression (4) is the formula for the Kendall rank correlation coefficient. Based on the hypotheses (H6), when K and L are counted, pairs whose HEs are in the inclusion relation are excluded. For HEs in E, if the ranks in the describing order are completely concordant with the ranks in the chronological order, $sc_{CO}(E)$ returns 1.

For measuring the geographical distance in the hypothesis (H7), some sort of geographical knowledge base is required. However, available geographical databases such as the GeoNames[5] are insufficient to support the geographical entities of world history because of countries that no longer exists and other inconsistencies. Therefore, we constructed a geographic thesaurus specialized in world history by extracting and clustering all geographical entities from the world history textbook published by Tokyo Shoseki.[6] The geographical entities are hierarchically grouped into classes of continent, subregion of continent, country and city. Using the geographic thesaurus, the geographical ordering score sc_{GO} is calculated by the following expression.

$$sc_{GO}(E) = \frac{1}{geochange(E) + 1} \tag{5}$$

$$geochange(E) = \frac{1}{|E| - 1} \sum_{i=1}^{|E|-1} distance(range(e_i), range(e_{i+1})) \tag{6}$$

where $range(e)$ is a function to return a thesaurus node that is the nearest common node subsuming all geographical entities included in the HE e, and $distance(n_i, n_j)$ is a function to return the shortest distance between the thesaurus nodes n_i and n_j. If there is no change in the range of HEs in E, $sc_{GO}(E)$ returns 1.

4.2.2 *Cooperability Score.*

Based on the hypothesis (H8), the chronological cooperability score sc_{CC} is calculated by the following expression.

$$sc_{CC}(E, CC) = \frac{overlap(period(E), CC)}{extend(period(E), CC)} \tag{7}$$

where $period(E)$ is a function to return a pair of the earliest time and the latest time in E, $overlap(P_1, P_2)$ is a function to return the length of the overlap period between P_1 and P_2, and $extend(P_1, P_2)$ is a function to return the length of the period between the earliest time and the latest time among P_1 and P_2. Note that $period()$ deals with the times that can determine the end of the period. If there are two periods of HEs "from 1900 A.D. to 1910 A.D." and "up to 1920 A.D.," $period()$ returns the period "from 1900 A.D. to 1920 A.D." although the later may be occurred before 1900 A.D. When the period of E is exactly overlapped the period of CC, $sc_{CC}(E, CC)$ returns 1.

[5] http://www.geonames.org/

[6] http://www.tokyo-shoseki.co.jp/ (in Japanese)

Based on the hypothesis (H9), the geographical cooperability score sc_{GC} is calculated by the following expression.

$$\begin{aligned} sc_{GC}(E,GC) &= \frac{2P(E,GC)R(E,GC)}{P(E,GC)+R(E,GC)} && (8) \\ P(E,GC) &= \frac{subsumed(geoentities(E),GC)}{|geoentities(E)|} && (9) \\ R(E,GC) &= \frac{subsuming(geoentities(E),GC)}{|GC|} && (10) \end{aligned}$$

where $geoentities(E)$ is a function that returns a set of geographical entities included in E, $subsumed(G_1,G_2)$ is a function that returns the number of geographical entities of G_1 subsumed by geographical entities of G_2, and $subsuming(G_1,G_2)$ is a function that returns the number of geographical entities of G_2 subsuming geographical entities of G_1. The expression (8) is the harmonic mean of precision and recall between the geographical entity set of E and GC. If all geographical entities of E are subsumed under GC and all geographical entities of GC subsume at least one of the geographical entities of E, $sc_{GC}(E,GC)$ returns 1.

4.2.3 Uniformity Score.

While there is always something described in a GS, the description does not always correspond to a particular category of the geographic thesaurus, such as a country. We used the standard deviation of the depth of category nodes in the geographic thesaurus for the geographical uniformity, while information entropy is used for the quantity uniformity. Based on the hypothesis (H4), the geographical uniformity score sc_{GU} is calculated by the following expression.

$$\begin{aligned} sc_{GU}(SS) &= 1-\frac{sd_{GU}(S)}{am_{GU}(SS)} && (11) \\ sd_{GU}(SS) &= \sqrt{\frac{1}{|SS|}\sum_{i=1}^{|SS|}(depth(s_i)-am_{GU}(SS))^2} && (12) \\ am_{GU}(SS) &= \frac{1}{|SS|}\sum_{i=1}^{|SS|}depth(s_i) && (13) \end{aligned}$$

where $depth(s)$ is a function to return the distance between the thesaurus root node and the node corresponding to the range of s. When all depths the ranges of GSs in SS, $sc_{GU}(SS)$ returns 1.

Based on the hypothesis (H5), the quantity uniformity score sc_{QU} is calculated by the following expression.

$$\begin{aligned} sc_{QU}(SS) &= \frac{-\sum_{i=1}^{|SS|}p(s_i,SS)\log_2 p(s_i,SS)}{\log_2|SS|} && (14) \\ p(s,SS) &= \frac{length(s)}{\sum_{i=1}^{|SS|}length(s_i)} && (15) \end{aligned}$$

where $length(s)$ is a function to return the number of characters described in s. The expression (14) is the normalized formula for information entropy. When all numbers of characters in GSs of SS are equal, $sc_{QU}(SS)$ returns 1.

5 EXPERIMENTAL RESULT

Using all essays submitted to the QA Lab-2 Phase-1 and -3 [11], we compared the scores measured by the proposed method and the scores evaluated by human expert. Although the number of the essays is only 15, they are annotated with the marks granted and taken away besides the total score by a human expert. Note that the essays are mixed with essays answering 8 different questions. Basically the marks awarded take account of the correctness of the content, and the marks lost account for the ill-formedness. With this, we compared the scores to the method behind subtracting marks. Note that the lost marks are caused by not only chronological and geographical inconsistencies.

Figure 4 shows the scatter plot between the scores by our method and the subtracted marks. The two dots in the circle of Figure 4 are far apart. They represent the essays answering the same question, and the other dots are essays answering the other questions. The question of the two essays asks for an overview of Egyptian history since the birth of Egyptian civilization. The chronological condition is helpless to screen HEs chronologically, and the geographical condition is limited to Egypt - a relatively small region. In this case, almost all HEs satisfy the chronological condition, and the GS structure is flat, which means there is only a single (root) GS and there is no sub-GS. As a result, the method scores are extremely high as long as the essays describe the HEs in Egypt in chronological order. Except for two essays, the correlation coefficient was 0.21, which indicated a weak positive correlation. Taking into account that the marks subtracted include other causes than the chronological and geographical problems, the value seems to be fairly good. However, the sample size was small and there is much room for improvement of the method. We will conduct further research with a larger number of essays.

6 RELATED WORK

The linguistic well-formedness in the DUC workshop and the readability/fluency in the TAC Guided Summarization tasks were evaluated in terms of grammaticality, non-redundancy, referential clarity, focus, and 'structure and coherence'. Our measures are relative to the focus and 'structure and coherence'.

Although Barzilay et al. [2] and Okazaki et al. [9] researched the chronological ordering, they did not take account of geographical information. Buscaldi et al. [4] found that geography is related to semantic similarity, but they only aimed to measure semantic equivalence between two text snippets. Because Madanani et al. [7] only researched sentence ordering, the research only applied to the context of a short, domain-independent summarization. Bauer and Teufe [3] proposed the extended Pyramid method for timeline summarization, but they did not focus on the well-formedness. Although Wagner et al. [13] researched the well-formedness, they focused only on grammatical errors. Therefore, there is no research on a methodology for measuring the focus and the structure and coherence of world history essays in terms of the chronological and geographical information.

7 CONCLUSION

For world history essays in Japanese university entrance exams, we proposed a method for measuring the uniformity, ordering and cooperability in terms of the chronological and the geographical information. The features of well-formedness are found by observing several model answer essays. From the experimental result, we found a weak positive correlation between the scores measured by

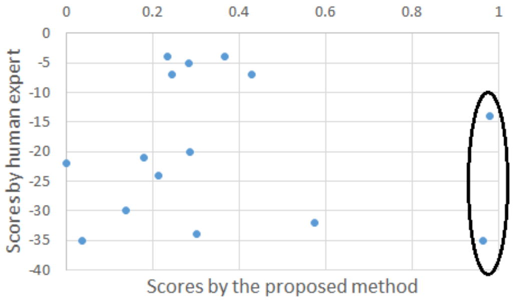


Figure 4: The scatter plot between the scores by the method and the scores by a human expert

our method and the scores estimated by a human expert in world history. The scoring functions of the method are based on simple concepts. We will investigate more appropriate functions in the future.

REFERENCES

[1] Eugene Agichtein, David Carmel, Donna Harman, Dan Pelleg, and Yuval Pinter. 2015. Overview of the TREC 2015 LiveQA Track. In *Proceedings of The Twenty-Fourth Text REtrieval Conference.*

[2] Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. 2002. Inferring Strategies for Sentence Ordering in Multidocument News Summarization. *Journal of Artificial Intelligence Research* 17, 1 (2002), 35–55.

[3] Sandro Bauer and Simone Teufe. 2015. Improving Chronological Sentence Ordering by Precedence Relation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, Vol. 2. 834–839.

[4] Davide Buscaldi, Jorge J. Garcia Flores, Joseph Le Roux, and Nadi Tomeh. 2014. LIPN: Introducing a new Geographical Context Similarity Measure and a Statistical Similarity Measure Based on the Bhattacharyya Coefficient. In *Proceedings of the 8th International Workshop on Semantic Evaluation.* 400–405.

[5] Herbert P. Grice. 1975. Logic and Conversation. In *Syntax and Semantics: Vol. 3: Speech Acts*, P. Cole and J. L. Morgan (Eds.). Academic Press, San Diego, CA, 41–58.

[6] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of Workshop on Text Summarization Branches Out.* 74–81.

[7] Nitin Madnani, Rebecca Passonneau, Necip Fazil Ayan, John M. Conroy, Bonnie J. Dorr, Judith L. Klavans, Dianne P. O'Leary, and Judith D. Schlesinger. 2007. Measuring Variability in Sentence Ordering for News Summarization. In *Proceedings of the Eleventh European Workshop on Natural Language Generation.* 81–88.

[8] Ani Nenkova and Rebecca J. Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics.* 145–152.

[9] Naoaki Okazaki, Yutaka Matsuo, and Mitsuru Ishizuka. 2004. Improving Chronological Sentence Ordering by Precedence Relation. In *Proceedings of the 20th International Conference on Computational Linguistics.* 81–88.

[10] Rebecca J. Passonneau, Emily Chen, Weiwei Guo, and Dolores Perin. 2013. Automated Pyramid Scoring of Summaries using Distributional Semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics.* 143–147.

[11] Hideyuki Shibuki, Kotaro Sakamoto, Madoka Ishioroshi, Akira Fujita, Yoshinobu Kano, Teruko Mitamura, Tatsunori Mori, and Noriko Kando. 2016. Overview of the NTCIR-12 QA Lab-2 Task. In *Proceedings of The NTCIR-12 Conference.*

[12] Hideyuki Shibuki, Kotaro Sakamoto, Yoshinobu Kano, Teruko Mitamura, Madoka Ishioroshi, Kelly Y. Itakura, Di Wang, Tatsunori Mori, and Noriko Kando. 2014. Overview of the NTCIR-11 QA-Lab Task. In *Proceedings of The NTCIR-11 Conference.*

[13] Joachim Wagner, Jennifer Foster, and Josef van Genabith. 2007. A Comparative Evaluation of Deep and Shallow Approaches to the Automatic Detection of Common Grammatical Errors. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.* 112–121.

Video Question Answering to Find a Desired Video Segment

Mayu Otani
Nara Institute of Science and Technology
otani.mayu.ob9@is.naist.jp

Yuta Nakashima
Osaka University
n-yuta@ids.osaka-u.ac.jp

Esa Rahtu
University of Oulu
esa.rahtu@ee.oulu.fi

Janne Heikkiä
University of Oulu
janne.heikkila@ee.oulu.fi

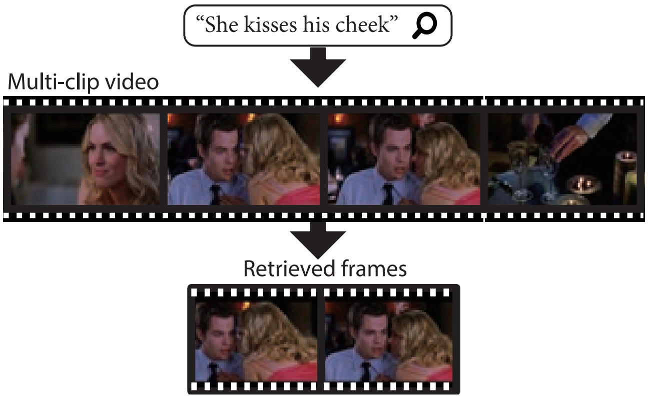


Figure 1: The FGVR task finds specific segments in a long video that matches a natural language query.

ABSTRACT

Fine-grained video retrieval (FGVR) is a technique for finding a segment in a long video using a natural language query provided by the user. In this demo, we extend FGVR to a simple question answering system to find if a given video clip contains a desired segment and ground it by showing the segment.

KEYWORDS

Fine-grained video retrieval, deep neural network, question answering

1 INTRODUCTION

Content-based video retrieval is one of the widely studied topics, and recent deep neural networks (DNNs) have enabled us to do this using natural language queries without relying on metadata assigned to each video in a database by, *e.g.*, mapping a video and natural language queries into the same semantic space [4, 5]. Such approaches mainly deal with video clips that may only contain a single event or action. However, most videos are edited and consist of multiple video clips (*e.g.*, movies, TV programs, and YouTube videos) or lengthy and unedited (*e.g.*, surveillance video); therefore, more realistic video retrieval applications may involve finding one or more segments (in different lengths) that match the query in a long, multi-clip video. One example of such applications can be rapidly finding a specific scene in a movie or identifying a certain event in a surveillance video.

We refer to the task of finding one or more video segments in a video clip to *fine-grained video retrieval*, or *FGVR* in short (Figure 1). Various approaches can address this task. For example, existing video retrieval approaches [4, 5] that deal with short video clips can be applied by segmenting a long video into shorter ones, which may require sophisticated video segmentation or lose temporal dependencies among different segments. Another interesting approach can be judging if a frame matches the query or not with retaining temporal dependencies by using recurrent neural networks, which we call the *frame-level* approach.

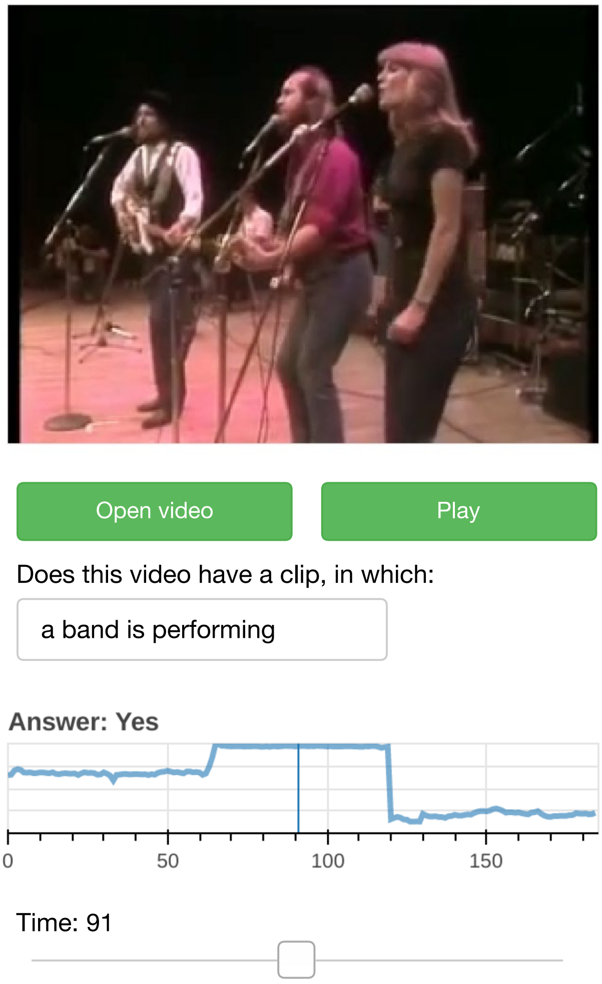
In this demo, we extend the idea of FGVR to a question answering system that firstly answers to the question in a specific form (*i.e.*, "Does this video contain a clip, in which ...") and show a corresponding clip for grounding. We implement a DNN-based system in the frame-level approach. One practical problem to realize this system is the lack of a dataset to train the DNN. We address this problem by concatenating randomly selected short video clips, which allows us to generate an arbitrary number of long videos with corresponding natural language queries.

2 DEMO SYSTEM OVERVIEW

Figure 2 shows the screenshot of our demo system. The top pane shows the video to be retrieved. "Open video" and "Play" buttons are to load the video to be retrieved and to play it back. Below these buttons is the text box to specify the question. The answer to the question (either "Yes" or "No") is shown below. The graph shows the frame-level relevance between the question (or the text in the text box) and the video. If the video has frames with relevance scores higher than a predetermined threshold, the system set the answer to "Yes." Using the slider at the bottom, the user can freely browse the video. In the demo, users can try some multi-clip videos synthesized based on YouTube videos in the Microsoft Video to Text dataset [3] as well as movies from MPII Video Description datasets [2].

3 DNN-BASED FGVR

The key component of our demo system is DNN-based FVGR in the frame-level approach, that computes the frame-level relevance scores given a video and a natural language query. Figure 3 shows the network architecture. After the user specifies the video to be retrieved and inputs the question (or the query) in the text box, video X is decomposed into a sequence $(x_1, \dots, x_T)$ of frames x_t, where each frame is transformed into a feature vectors $V = (v_1, \dots, v_T)$ using ResNet [1], and the query Y is decomposed into a sequence $(y_1, \dots, y_M)$ of words y_m.



Does this video have a clip, in which:

a band is performing

Answer: Yes

Time: 91

Figure 2: A screenshot of our demo system.

The feature vector v_t in V computed from video frame x_t is fed into bidirectional LSTM layers, which produce two hidden states for time step t. These hidden states are concatenated into a single vector and passed to a two-layer perceptron with the hyperbolic tangent nonlinearity to obtain the video encoding for this time step. Due to the bidirectional LSTM layers, the video embedding for each time step can contain temporal dependencies to describe the concept included in the nearby frames. For the word sequence $(y_1, \ldots, y_M)$ obtained from the query, each word y_m is transformed into word vector and then a single LSTM layer is used to generate a query embedding. The video embedding and the query embedding have the same dimensionality (*i.e.*, 256-D) so that the relevance score between them can be computed using the cosine similarity function.

Since this is a very new task, there is no dataset that can be used for training this DNN. Therefore, we automatically synthesize multi-clip video and natural language query pairs based on existing datasets for video captioning (*i.e.*, YouTube videos [3] and movies [2]). Firstly we pick out a single video clip in a dataset together with its corresponding caption, and then randomly pick out other two video clips in the dataset. These three videos are randomly shuffled and concatenated into a longer video clip. We use these data to train the DNN.

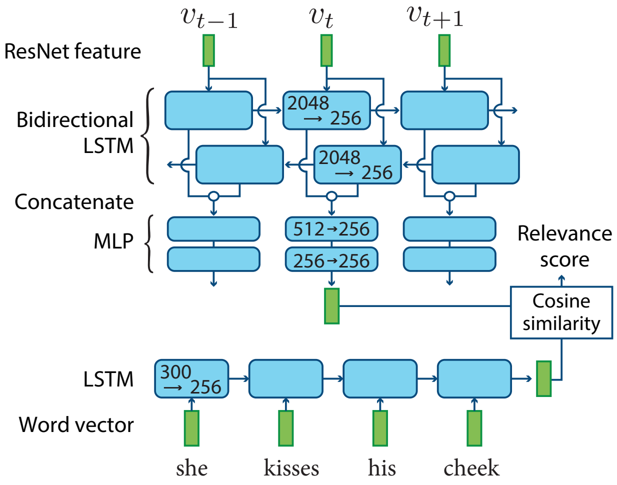


Figure 3: Bi-LSTM network architecture for computing relevance scores.

4 CONCLUSION

In this demo, we show how our question answering system over DNN-based FGVR works. This task can be the basis for various types of video retrieval applications, such as movie scene identification and event extraction in a surveillance video. The DNN of our current implementation is relatively simple but shows promising performance. Our next step is to evaluate our system in a more realistic scenario (*e.g.*, movie scene identification), which requires making a dataset by human annotators. This work is partly supported by JSPS KAKENHI No. 16K16086.

REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. 770–778.
[2] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. 2017. Movie Description. *IJCV* 123, 1 (2017), 94–120.
[3] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*. 5288–5296.
[4] R Xu, C Xiong, W Chen, and JJ Corso. 2015. Jointly Modeling Deep Video and Compositional Text to Bridge Vision and Language in a Unified Framework. In *AAAI*. 2346–2352.
[5] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. 2016. End-to-end Concept Word Detection for Video Captioning, Retrieval, and Question Answering. *arXiv preprint, arXiv:1610.02947* (2016), 20 pages.

OKBQA Framework for collaboration on developing natural language question answering systems

Prototype System Demonstration

Jin-Dong Kim
DBCLS and KAIST
jdkim@dbcls.rois.ac.jp

Christina Unger
University of Bielefeld
cunger@cit-ec.uni-bielefeld.de

Axel-Cyrille Ngonga Ngomo
University of Paderborn
axel.ngonga@upb.de

André Freitas
University of Passau
andrenfreitas@gmail.com

Young-gyun Hahm
KAIST
hahmyg@kaist.ac.kr

Jiseong Kim
KAIST
jiseong@kaist.ac.kr

Sangha Nam
KAIST
nam.sangha@kaist.ac.kr

Gyu-Hyun Choi
KAIST
wiany11@kaist.ac.kr

Jeong-uk Kim
KAIST
kju0627@gmail.com

Ricardo Usbeck
University of Paderborn
ricardousbeck@gmail.com

Myoung-Gu Kang
Young Plus Soft Corp.
mgkaki@youngplussoft.com

Key-Sun Choi
KAIST
kschoi@kaist.ac.kr

ABSTRACT

The OKBQA Framework is developed to facilitate an open collaboration for development of natural language question-answering systems. It defines necessary modules with their API, so that OKBQA-conformant modules can inter-operate with each other. The OKBQA repository (http://repository.okbqa.org) is where those modules are registered, and the OKBQA demo system (http://ws.okbqa.org/wui-2016/) allows composition and execution of workflows using the modules.

KEYWORDS

question-answering, natural language processing, collaboration platform, SPARQL generation

1 INTRODUCTION

The OKBQA framework (http://www.okbqa.org) has been developed as a platform for open collaboration on development of natural language (NL) question-answering (QA) systems. With the goal to facilitate collaboration through distributed voluntary contributions, activities around the framework include (1) identifying and defining modules necessary for NLQA, and their APIs, (2) implementing them, and (3) developing and maintaining a public service whereon workflows of QA can be composed and executed. Recently, the development has reached a milestone: a demo system has begun to work. On the system, workflows for NLQA can be easily composed and executed, using modules which are deployed as REST services. To demonstrate the functionality of the framework, two workflows for QA in English and Korean have been set up on the system.

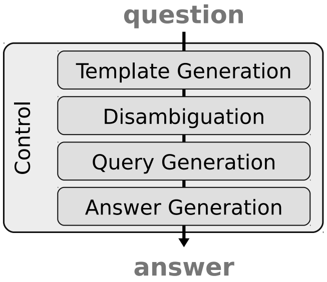
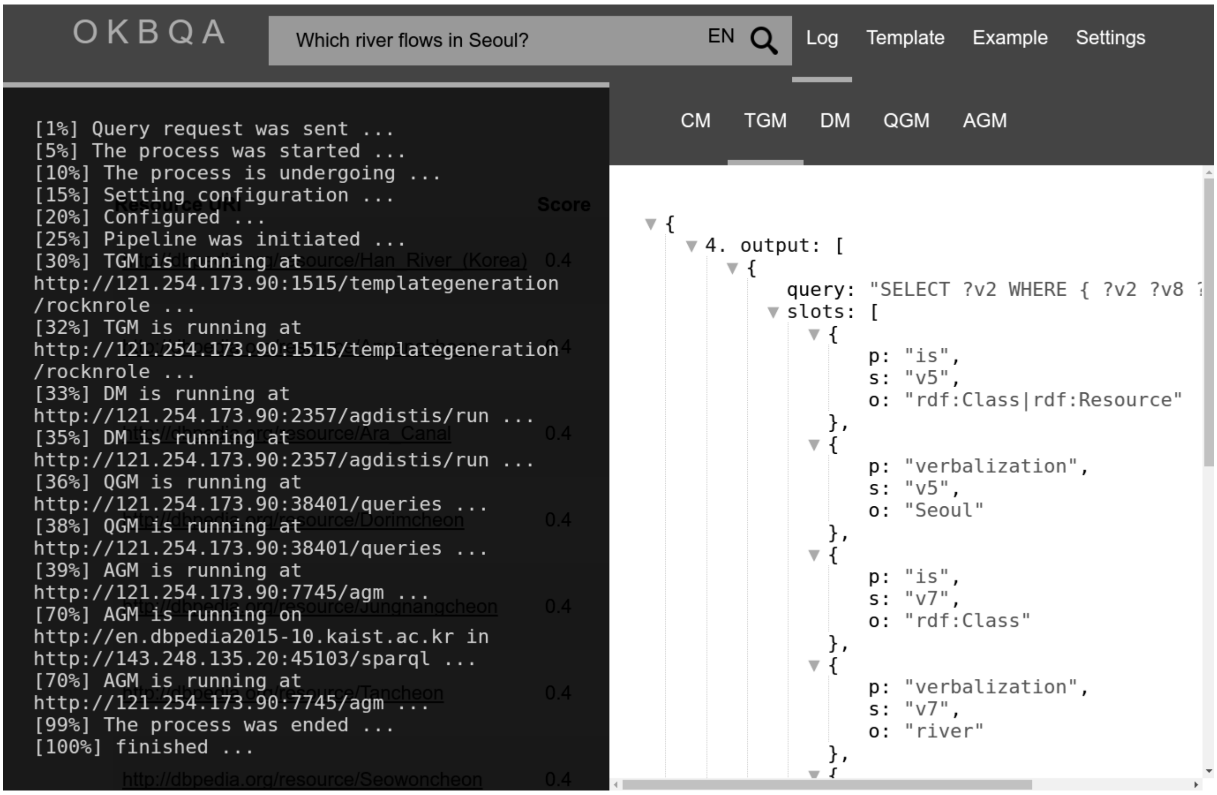


Figure 1: OKBQA modules in a model flow

2 OKBQA FRAMEWORK

Figure 1 gives an overview of the OKBQA framework. In its modular architecture, it currently defines four core categories of modules: *Template Generation Module (TGM)*, *Disambiguation Module (DM)*, *Query Generation Module (QGM)*, and *Answer Generation Module (AGM)*. A *Controller Module (CM)* is supposed to make a workflow of QA by connecting several core modules. The input of a core workflow is supposed to be a natural language query in character string, and the output to be a list of URIs or literals.

Two design choices were made to ease collaboration among different groups: (1) each module needs to be accessible as a REST service, and (2) the input and output of each module are represented in JSON. Due to the design, a module can be implemented in any programming language, and it can be deployed to any location in the net. A workflow is then defined as a sequence of REST services, which makes it easy to compose a workflow using modules distributed in the net.



```
[1%] Query request was sent ...
[5%] The process was started ...
[10%] The process is undergoing ...
[15%] Setting configuration ...
[20%] Configured ...
[25%] Pipeline was initiated ...
[30%] TGM is running at
http://121.254.173.90:1515/templategeneration
/rocknrole ...
[32%] TGM is running at
http://121.254.173.90:1515/templategeneration
/rocknrole ...
[33%] DM is running at
http://121.254.173.90:2357/agdistis/run ...
[35%] DM is running at
http://121.254.173.90:2357/agdistis/run ...
[36%] QGM is running at
http://121.254.173.90:38401/queries ...
[38%] QGM is running at
http://121.254.173.90:38401/queries ...
[39%] AGM is running at
http://121.254.173.90:7745/agm ...
[70%] AGM is running on
http://en.dbpedia2015-10.kaist.ac.kr in
http://143.248.135.20:45103/sparql ...
[70%] AGM is running at
http://121.254.173.90:7745/agm ...
[99%] The process was ended ...
[100%] finished ...
```

```
▼{
  ▼4. output: [
    ▼{
        query: "SELECT ?v2 WHERE { ?v2 ?v8 ?
      ▼slots: [
        ▼{
            p: "is",
            s: "v5",
            o: "rdf:Class|rdf:Resource"
          },
        ▼{
            p: "verbalization",
            s: "v5",
            o: "Seoul"
          },
        ▼{
            p: "is",
            s: "v7",
            o: "rdf:Class"
          },
        ▼{
            p: "verbalization",
            s: "v7",
            o: "river"
          },
```

Figure 2: A screenshot of the OKBQA prototype demo system

3 REPOSITORY

The OKBQA repository is maintained to provide a venue for storing information about modules developed for the OKBQA framework (http://repository.okbqa.org). At the time of writing, there are 24 modules registered to the repository, which include a TGM module from the AutoSPARQL project [2], a DM based on AGDISTIS [3], a QGM from the LODQA project [1], and several modules from the ExBrain project (http://exobrain.kr/).

4 DEMO SYSTEM

A prototype demo system is developed and maintained as a public service (http://ws.okbqa.org/wui-2016/), to demonstrate how workflows in OKBQA actually work, and also to support development of new modules. Currently, two workflows have been set-up for QA in English and Korean. Users can choose a workflow and try it with NL queries. An important point here is that the system will show not only the final results but also the output of each module. Figure 2 shows a screen-shot of the system with an example query *Which river flows in Seoul?*. During execution, it shows the progress of the workflow in the left pane, and the output of each module on the right. Through the interface, users can inspect how each module of the workflow works. For those who are new to the framework, such an interface may give a chance to figure out how an OKBQA workflow works. More importantly, The system allows users to freely modify a workflow by replacing a module with a new one. A newly developed module, once it is deployed as a REST service, can be plugged-in to a workflow. By inspecting its IO in the workflow, the developer may also be able to figure out how it works in the workflow. In this way, the prototype demo system is designed to support the development of modules for OKBQA.

5 CONCLUSION

For those who are interested in developing NLQA systems, we expect the resources of OKBQA to provide a good starting point. The system just began to work and there is a large room for improvement. For example, the composition of a workflow is not yet sufficiently flexible, and the performance of current reference workflows is not yet competitive. Nevertheless, we believe it is a significant milestone that such a framework has begun to work to organize contributions by different groups. We hope this presentation to be an opportunity to receive feedback from interested parties and also to invite potential collaborators.

REFERENCES

[1] Jin-Dong Kim and K Bretonnel Cohen. 2013. Natural language query processing for SPARQL generation: A prototype system for SNOMED-CT. In *Proceedings of BioLink SIG meeting 2013*.

[2] Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. 2012. Template-based Question Answering over RDF Data. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12)*. ACM, New York, NY, USA, 639–648.

[3] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Sören Auer, Daniel Gerber, and Andreas Both. 2014. AGDISTIS - Agnostic Disambiguation of Named Entities Using Linked Open Data. In *European Conference on Artificial Intelligence*.

FelisCatusZero: A world history essay question answering system for the University of Tokyo's entrance exam

Kotaro Sakamoto
Yokohama National University
National Institute of Informatics
sakamoto@forest.eis.ynu.ac.jp

Takaaki Matsumoto
Carnegie Mellon University
tmatsumo@andrew.cmu.edu

Madoka Ishioroshi
National Institute of Informatics
ishioroshi@nii.ac.jp

Hideyuki Shibuki
Yokohama National University
shib@forest.eis.ynu.ac.jp

Tatsunori Mori
Yokohama National University
mori@forest.eis.ynu.ac.jp

Noriko Kando
National Institute of Informatics
The Graduate University for
Advanced Studies (SOKENDAI)
kando@nii.ac.jp

Teruko Mitamura
Carnegie Mellon University
teruko@cs.cmu.edu

ABSTRACT

FelisCatusZero is an open-source system to answer world history essay questions of the University of Tokyo's entrance exam in Japanese, which is based on extractive multi-document summarization.

KEYWORDS

Question Answering, Essay, World History, Japanese University Entrance Examination

1 INTRODUCTION

Question answering is widely noticed as an information access technology to answer various complex information needs. However, many previous researches of question answering have focused on comparatively simple questions, such as "Who is Donald Trump?". The real world questions sometimes differ from the previous researches in writing long background/circumstances before the question focus or including abstractive expressions. As the first step of tackling the real world questions, we are investigating question answering technologies of a university entrance exam's world history subject[1][2]. The University of Tokyo's entrance exam of world history has questions that a student has to write an essay with no more than 400 to 600 Japanese characters. Fig 1 shows an English translation of a question from the University of Tokyo's entrance exam. Fig 2 shows an English translation of a gold standard. In this paper, we present FelisCatusZero[1] which answers the essay questions. The system's essay answer generating algorithm is based on extractive multi-document summarization. We tested the system on a few mock exams of the University of Tokyo's entrance exam in Japanese. The results showed that the system got scores close to the average scores of students who were preparing for the exam.

> We are currently living in an era of information revolution, and the pace of globalization is accelerating. Not only people and goods are flowing across oceans and borders with increasing frequency; information is being transmitted across the world almost in real time. Underlying these developments is the rapid progress that has been made in transportation and communication technologies.
> · · ·
> Furthermore, these technological innovations are noteworthy for the parts they played in Western nations' invasions of Asia and Africa. For example, the Reuters news agency gathered information from around the world to help develop the international presence of the British Empire. But, on the other hand, global information sharing and accelerated migration facilitated by the development of transportation were also stimulating factors in the growth of local nationalism.
>
> Write an essay explaining, in 225 English words (550 Japanese characters) or less, how developments in the means of transportation and communication prompted the colonization of Asia and Africa and heightened local nationalism. Use all nine keywords shown below at least once.
> Suez Canal, steamship, Baghdad Railway, Morse code, Marconi, the Boxers, Russo-Japanese War, Persian Constitutional Revolution, Gandhi

Figure 1: A translation of an example of an essay question

> · · · These technologies were used in Western advances into Asia and Africa. For example, the opening of the Suez Canal greatly reduced travel time between Europe and Asia, but the British controlled the canal to maintain a route to India. With regard to African policy, Britain supported Cecil Rhodes, who advocated a plan for connecting Cape Town to Cairo via rail and telegraph. Germany implemented a 3B policy, which included the construction of the Baghdad Railway, to advance into the Middle East. · · ·

Figure 2: A translation of an example of the gold standard

[1] github.com/ktr-skmt/FelisCatusZero-multilingual/

2 KNOWLEDGE SOURCES

We use four high school textbooks and one glossary of world history subject as knowledge sources. Fig 3 shows an example of a textbook. Fig 4 shows an example of the glossary.

> <DOC> <DOCNO>Y-JH-14-01-5</DOCNO>
> <TITLE>Imperialism and Asian nationalist movements-Imperialism and powerful imperialist countries' deployments-Russia</TITLE>
> <TEXT>Since the 1890s, Russian capitalism has developed through the capital import from France, and great industries have grown rapidly in cities. · · · </TEXT> </DOC>

Figure 3: A translation of a paragraph example of a textbook

> <DOC> <DOCNO>YamakawaWorldHistoryGlossary-5290</DOCNO>
> <TITLE>Russia-Japan agreement</TITLE>
> <TEXT>An agreement with the intention of the corporate protection of the interests of both Japan and Russia after the Russo-Japanese War. · · · </TEXT> </DOC>

Figure 4: A translation of an entry example e of the glossary

3 OUTLINE

Fig 5 shows the outline of FelisCatusZero. We input the question as an XML file. The system extracts the keywords which have to be included in the answer, the character limit and the time range from the question. The system retrieves documents including any keywords from the knowledge sources, extracts and groups sentences by each keyword. The system ranks the sentences based on the scores which mean to what extent they should be contained in answer. To generate an answer candidate, the system selects a sentence from each sentence group and sorts the selected sentences chronologically. The system removes answer candidates if they exceed the character limit. The system ranks the answer candidates by the summation of sentence scores. The system chooses an answer candidate with the highest score as the final answer. Finally the system outputs the answer as an XML file.

4 EXPERIMENT

We have let the system answer a question of a mock exam for the University of Tokyo's entrance exam four times so far. Every time the system answered a question, a human expert in world history evaluated the system answer and gave the system some minor updates.

4.1 Results

Fig 6 shows translations of an example of the system answer. Table 1 shows the four-time expert evaluation results.[2] [3]

Q	SCORE	AVE. SCORE OF STUDENTS	
1st	4	7.3	out of 28
2nd	9	4.3	out of 26
3rd	3	4.6	out of 20
4th	10	9.2	out of 20

Table 1: Evaluation results

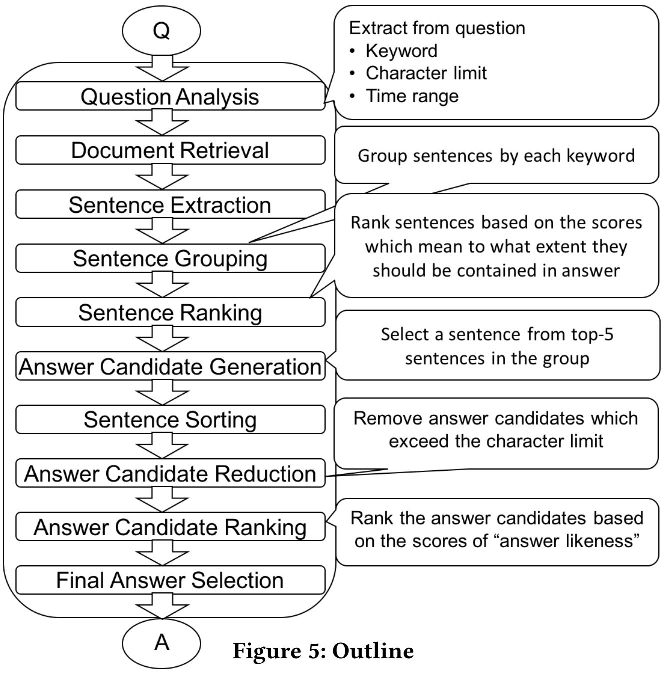

Q
Question Analysis
Document Retrieval
Sentence Extraction
Sentence Grouping
Sentence Ranking
Answer Candidate Generation
Sentence Sorting
Answer Candidate Reduction
Answer Candidate Ranking
Final Answer Selection
A
Extract from question
• Keyword
• Character limit
• Time range
Group sentences by each keyword
Rank sentences based on the scores which mean to what extent they should be contained in answer
Select a sentence from top-5 sentences in the group
Remove answer candidates which exceed the character limit
Rank the answer candidates based on the scores of "answer likeness"

Figure 5: Outline

4.2 Discussions

The system can select many relevant sentences from knowledge sources, though the retrieval accuracy is not good enough to get the perfect score. Also, we think that the system needs an essay structure correction.

> · · · The opening of the Suez canal in 1869 encouraged the advancement of the Great powers into Africa, which became the target of partitioning disputes. The Baghdad railway was targeted by many countries, but Germany obtained the rights of construction in 1899, and in 1903 it founded a society. In the end, the Baghdad railway was only partially opened.

Figure 6: A translation of an example of the system answer

5 CONCLUSIONS

We presented an open-source system FelisCatusZero to answer world history essay questions of the University of Tokyo's entrance exam in Japanese. It is based on extractive multi-document summarization. We tested the system on a few mock exams of the University of Tokyo's entrance exam in Japanese. The results showed that the system got scores close to the average score of students who were preparing for the exam. However, to get a higher score, we still have much room for improvement, not limited to accuracies of sentence extraction and extracted sentence combination.

REFERENCES

[1] Kotaro Sakamoto, Madoka Ishioroshi, Hyogo Matsui, Takahisa Jin, Fuyuki Wada, Shu Nakayama, Hideyuki Shibuki, Tatsunori Mori, and Noriko Kando. 2016. Overview of the NTCIR-12 QA Lab-2 Task. In *Proceedings of The NTCIR-12 Conference.*

[2] Hideyuki Shibuki, Kotaro Sakamoto, Madoka Ishioroshi, Akira Fujita, Yoshinobu Kano, Teruko Mitamura, Tatsunori Mori, and Noriko Kando. 2016. Overview of the NTCIR-12 QA Lab-2 Task. In *Proceedings of The NTCIR-12 Conference.*

[2] Note that the system could not extract time range from the first question due to an XML tagging error.

[3] In the third question, the system could get 6 by a minor update.

Index of Authors