NLP-NITMZ @ MSIR 2016 System for Code–Mixed Cross–Script Question Classification

Goutam Majumder
National Institute of Technology Mizoram
Deptt. of Computer Science & Engg.
Mizoram, India
goutam.nita@gmail.com

Partha Pakray
National Institute of Technology Mizoram
Deptt. of Computer Science & Engg.
Mizoram, India
parthapakray@gmail.com

ABSTRACT

This paper describes our approach on Code–Mixed Cross–Script Question Classification task, which is a subtask 1 of MSIR 2016. MSIR is a Mixed Script Information Retrieval event in conjunction with FIRE 2016, which is the *8th meeting of Forum for Information Retrieval Evaluation*. For this task, our team **NLP–NITMZ** submitted three system runs such as: i) using a *direct* feature set; ii) using *direct* and *dependent* feature set and iii) using Naive Bayes classifier. The first system is our baseline system, which is based *direct* feature sets and we used a group of keywords to generate this direct feature set. To identify question classes our baseline system falls in ambiguity (means one question is tagged with multiple classes). To deal with this ambiguity, we developed another set of feature and we consider this feature set as *dependent* feature set, because *keywords* from this set is worked with *direct* feature set. The highest accuracy of our system is 78.88% using method–2 and we submitted as run–3. Our other two runs have same accuracy as 74.44%.

Keywords

Natural Language Processing; Question Answering; Information Retrieval; Question Analysis

1. INTRODUCTION

Question Answering (QA) concerned with the building system, which can answer the questions automatically posed by human. The QA is a common discipline within the fields of Information Retrieval (IR) and Natural Language Processing (NLP). It is a computer program, querying a structured or unstructured database of knowledge or information and constructs its answer [6].

The current research of QA deals with a wide range of question type such as fact-based, hypothetical, semantically constrained, and cross-lingual questions. The need and importance of a QA system was first introduced in 1999, by the first QA task in TREC 8 (Text REtrieval Conference). It was revealed the need of a sophisticated search engines, which is able to retrieve the specific piece of information that could be considered as the best possible answer to a user question. In present days such QA system works as a backbone for successful of any E-Commerce business. In this type of systems, many frequently asked question (FAQ) files are generated based on most frequently asked use questions and types of those questions [4].

Being a classic application of NLP, QA has practical applications in various domains such as education, health care, personal assistance, etc. QA is a retrieval task, which is more challenging than the task of common search engine because the purpose of QA is to find out accurate and concise answer to a question rather than just retrieving relevant documents containing the answer [5].

Table 1: Example of Question Classes

Question Class	Examples
MNY	Airport theke Howrah Station volvo bus fare koto?
TMP	Volvo bus howrah station jete koto time nei?
DIST	Airport theke howrah station distance koto?
LOC	Airport theke textit kothai jabar bus nei?
ORG	Prepaid taxi counter naam ki?
OBJ	Murshidabad kon nodir tire obosthito?
NUM	Hazarduari te koto dorja ache?
PER	Ke Hazarduari toiri kore?
MISC	Early morning journey hole kon service valo?

In this paper, the participation of subtask 1 is reported, which is a code-mixed cross-script Question Classification task [2] of MSIR 2016 (Shared Task on Mixed Script Information Retrieval) [1]. The first step of understanding a question is to perform question analysis (QA). Question classification is an important task of QA, which detects the answer type of the question. Question classification not only helps to filter out a wide range of candidate answers but also determines answer selection strategies [3], [5].

In this subtask 1, given two sets as $Q = \{q_1, q_2, ..., q_n\}$ and $C = \{c_1, c_2, ..., c_n\}$, where Q be a set of factoid questions written in Romanized Bengali along with English and C be the set of question classes. The task is to classify each given questions into one of the predefined coarse-grained classes. Total of 9 question classes are given as classification task and example of each question class with specific tag is listed in Table 1.

Rest of the paper is organized as follows: in Section 2 we have discussed the three methods in detail. Performance of the three systems is analysed in Section 3 and we also compared with other submitted systems. Finally, conclusion of the report is drawn in Section 4.

2. THE PROPOSED METHOD

Three methods are developed for *MSIR16* subtask1 to identify the question classes. Two systems are based on feature sets and identification stages for questions are depen-

Table 2: Features of MNY Class with Example

Features	Questions as MNY Class
charge	Semi–official guide koto taka charge nei?
daam	Fuchhka r koto daam Darjeeling e?
price	Darjeeling e momo r price koto?
dam	Chicken momo r dam koto Darjeeling e?
khoroch	Pykare te boating e koto khoroch hobe?
fee	Wasef Manzil er entry fee koto?
tax	Koto travel tax pore India border e?
pore	Koto travel tax pore India border e?
fare	Darjeeling e dedicated taxi fare koto?
taka	Digha te Veg meal koto taka?

Table 3: Features of DIST Class with Example

Features	Questions as DIST Class
distance	Kolkata theke bishnupur er distance koto?
duroto	Bangalore theke Ooty r by road duroto koto?
area	Ooty botanical garden er area koto hobe?
height	Susunia Pahar er height koto hobe?
dure	Puri Bhubaneshwar theke koto dure?
uchute	Ooty sea level theke koto uchute?
km	Kolkata theke bishnupur koto km?

dent on these features. Two feature sets are identified first using the training dataset and we consider one set as *direct* and other set as *dependent*. For the third system we combined these two sets and machine learning features are build using Naive Bayes classifier. Details of the three methods are discussed next.

2.1 Method–1 (using direct feature set)

1. **MNY:** To identify the 'MNY' class 10 features/ keywords from the training dataset is identified and question contains these keywords are tagged as 'MNY' class. In Table 2, we have listed out all of these features with questions. With these 10 features we also identified another keyword *koto*, to tag questions as 'MNY' class. But it was analysed that, if we consider *koto* as *direct* feature for 'MNY' tag, then questions of other classes are also tagged as 'MNY' class. Such as '*Shankarpur Digha theke* ***koto*** *dure*?', which is a 'DIST' class type question.

   Like *koto*, *taka* feature is also unable to tag some 'MNY' questions. So we identified two other keywords as *dependent* feature set, which is discussed in Section 2.2.

2. **DIST:** Seven keywords are identified as *direct* feature set to tag the 'DIST' question class. We also consider same set of features for second method. All the identified keywords having the meaning as *distance* in Bengali as well as in English language. For this class no keyword is found for *dependent* set and in Table 3, we have listed out all the features with questions.

3. **TEMP:** Question contains any temporal unit such as *somoi*, *time*, *month*, *year* etc. are tagged as a **TEMP** class. For the temporal question class eight keywords are identified and all of these keywords are considered as *direct* feature set and no *dependent* features are considered. The *direct* set of features with examples are listed below:

   - **time**–Koto *time* lage Bangalore theke Ooty by road ?
   - **kobe**–*Kobe* Jorbangla Temple build kora hoyechilo ?
   - **kokhon**–Shyamrai Temple toiri hoi *kokhon* ?

Table 4: Features of ORG Class with Example

Features	Questions as ORG Class
ki	Prepaid taxi counter naam ki?
kara	World champion kara?
kon	kon team Ashes hereche?
ke	Ashes hereche ke?

4. **LOC:** To tag the location class only one *direct* feature as *kothai* is identified and this keyword is also used in second method. Examples for this class are given below:

   - **kothai** ras mela hoi ?
   - train r jonno **kothai** advice nite hobe ?

5. **ORG:** For organization class four *direct* features are identified and these features with questions are listed in Table 4. Among these four features, the *ki* feature has ambiguity with other question classes such as 'OBJ' and 'PER'. Examples of questions with multiple tags using *ki* feature is listed below:

   - **OBJ**–Ekhon **ki** museum hoye geche ?
   - **PER**–Rabindranath er babar naam **ki** ?

   The *kon* feature of 'ORG' class also having ambiguity issues with other classes such as 'TEMP' and 'OBJ'. Questions with *kon* feature of other classes are listed below:

   - **TEMP–Kon** month e vasanta utsob hoi shantiniketan e ?
   - **OBJ–Kon** mountain er upor Ooty ache ?

   Issues related to ambiguity are addressed in Section 2.2 with the help of *dependent* feature set.

6. **NUM:** Two direct features such as *koiti* and *koto* are identified to tag the questions as 'NUM' class. But *koto* keyword having ambiguity and questions of 'MNY' classes are tagged as 'NUM' . So a *dependent* feature set is identified, which merge with *koto* feature and this issues are discussed in Section 2.2. So in this method, to identified questions as 'NUM' we only consider the keyword *koiti* as feature.

   - Bishnupur e **koiti** gate ache ?
   - Leie **koiti** wicket niyeche ?

7. **PER:** To identify the 'PER' class five direct features are identified. Among these three features such as *ke*, *kake*, and *ki* are worked with *dependent* feature sets, which is discussed in Section 2.2 and two other features such as *kar* and *kader* consider for *direct* feature set. These *direct* features with example are listed bellow:

- **Kar** wall e sri krishna er life dekte paoya jabe ?
- Jagannath temple e **kader** dekha paben ?

8. **OBJ:** Two direct rules are found, but these rules are not able to identify the questions of 'OBJ' class. These rules are as follows:

   - Ekhon **ki** museum hoye geche ?
   - Hazarduari er opposit e **kon** masjid ache ?

   From these questions it is clearly understood that, the 'OBJ' class is in ambiguity with 'ORG' class. So these two features are used with other *dependent* features for question classification.

9. **MISC:** If no rules are satisfied then, questions are classified as 'MISC' class.

2.2 Method–2 (using direct and dependent feature set)

This *dependent* feature set is identified to improve the efficiency of the first method. The name of the second set is given as *dependent*, because some features of the *direct* sets are not able to identify the questions and those features work correctly when the *dependent* feature set is also available in the questions.

1. **MNY:** To identify the **MNY** class two *dependent* features such as *charge* and *koto* along with ten *direct* features are identified. If any questions contains any of these two features, it also look for the *direct* feature such as *taka* else it will not consider the 'MNY' class for this question and examples are listed below:

   - **koto**–Digha te Veg meal *koto taka* ?
   - **charge**–Semi-official guide koto *taka charge* nei ?

2. **DIST:** Same as method–1 using all set of *direct* features only.

3. **TEMP:** All direct features are used to tagged the 'TEMP' class.

4. **LOC:** No dependent features, same set of *direct* features of method–1 is used.

5. **ORG:** To handle the ambiguity issues with 'ORG' class questions three sets of *dependent* features are identified, which improves the system accuracy. In the first feature set, ambiguity with 'OBJ' class is addressed. By identifying a term such as *museum*, *mondir*, *mosque* which can qualify a question as 'OBJ' class. Examples of these features are as follows:

   - **museum**–ekhon ki *museum* hoye geche ?
   - **lake**–murshidabad e ki *lake* ache ?

   All such questions are not identified as 'ORG' class, instead of these questions are forwarded to the other feature set for prediction. In the second set, features are identified to handle the issues related to 'PER' class and the features are as follows:

   - **(*eche)**–ke *jiteche*, ke *hereche*, ke *hoyeche*
   - **team**–kon *team* Ashes *hereche* ?

Table 5: Ambiguity Classes with NUM Class

Ambiguity Classes	Questions in Ambiguity
DIST	Kolkata theke bishnupur er distance koto?
TEMP	Bishnupur e jete bus e koto time lagbe?
MNY	Indian citizen der entry fee koto taka?

   In questions, if tokens ends with the format *eche* and questions also have 'ORG' features then those questions are classified as 'PER' class. The third feature are identified not to deal with ambiguity, these set is worked with *kon* keyword of *direct* feature set used in method–1. In this set, we explicitly identified those words, which means an organization such as *shop*, *hotel*, *city*, *town* etc. and examples are listed below:

   - **shope**–*kon* **shop** e tea kena jete pare ?
   - **town**–rat 9 PM *kon* **town** ghumiye pore ?

6. **NUM:** The *koto* keyword of *direct* feature set for 'NUM' class, is also in ambiguity with other classes such as 'DIST', 'TEMP', and 'MNY'. So the *direct* features are not used here to tag the questions of **NUM** class, but are used to check rules, present in the questions or not, if yes then questions will not tag or else is tagged with 'NUM' class. Example of each ambiguity is listed in Table 5.

7. **PER:** We have used two *dependent* features to predict the questions of 'PER' class. In this method, *dependent* features are also worked with *direct* features for question prediction. Examples of questions of 'PER' class using *direct* and *dependent* features is listed below:

   - Chilka lake jaber tour conduct *ke* **kore** ?
   - Woakes *kake* run out **koreche** ?
   - Bangladesh r leading T20 **wicket-taker r** naam *ki* ?

8. **OBJ:** A *dependent* feature set is identified, which contains all such words those qualified as a object name to handle the ambiguity issues of 'OBJ' class with 'ORG' class. These object names are combined with two *direct* features such as *ki* and *kon*. Example of such ambiguity issues are listed below:

   - ekhon *ki* **museum** hoye geche ?
   - bengal r sobcheye boro **mosjid** *ki* ?
   - Nawab Wasef Ali Mirza r **residence** *ki* chilo ?

   From these *dependent* features, it was clear that, the *direct* features look for a token in the questions those have an entity of object type and these entities are represented as bold face in the examples. So for *kon direct* feature, same set of *dependent* features are used to identify the 'OBJ' class. Examples are as follows:

   - Berhampore-Lalgola Road e *kon* **mosjid** ache ?
   - Murshidabad kon **nodir** tire obosthito ?

9. **MISC** same as method–1.

Table 6: Statistics of Training Data set

Sl.No.	Question Class	Total
1	Money as **MNY**	26
2	Temporal as **TEMP**	61
3	Distance as **DIST**	24
4	Location as **LOC**	26
5	Object as **OBJ**	21
6	Organization as **ORG**	67
7	Number as **NUM**	45
8	Person as **PER**	55
9	Miscellaneous as **MISC**	5

2.3 Method–3 (Using Naive Bayes Classifier)

In this method, Naive Bayes classifier is used to train the model. For training, a feature matrix with probable class tags is input to the Bayes classifier. For each question in training set, one feature is considered and the last column of the feature matrix represents the question classes. This feature matrix is generated using the sets of *direct* and *dependent* features used in Method–1 and 2.

3. EXPERIMENT RESULTS

3.1 Data and Resources

Two datasets as training and testing data set are released for this task [1]. It was allowed that, participants can use any number of resource for this task. Each entry in dataset has the following format as *q_no q_string q_class* and is referred as *question_number*, *code–mixed cross–script question string* and *the class of the question* respectively. Training data set contains a total no. of 330 questions and is tagged among 9 question classes and details of the training data set with question classes is given in Table 6.

3.2 Results

For this task, NLP–NITMZ team submitted 3 system runs. Among these runs, two of them are feature/ keyword based and the third run is based on machine learning features. For the first run, different set of rules are identified for each question classes.

3.2.1 Run–1

The first run, is conducted using method–1, which is used all the *direct* rules. In this run, questions are identified and tagged with the class based on these *direct* rules. We achieved a success rate of 74.44% using Method–1 and the accuracy of identification of question classes are listed in Table 7 using the performance parameters.

3.2.2 Run–2

Naive Bayes classifier is used for this run. After being trained the model using training dataset, model is tested with the test data and class levels are predicted as classifier output. For this run it has the accuracy of 74.44%, which is same as run–1. In Table 3.2.2, the precision, recall and F–1 score for each class labels is listed.

3.2.3 Run–3

For this run the *direct* and *dependent* feature set is used together to address the ambiguity issues among the question classes. 78.89% is the success rate achieved in this run using Method–2 and the accuracy is listed in Table 3.2.3.

Table 7: Success rates using rules of Direct set

Classes	Precision	Recall	F–1 Score
MNY	0.80	0.50	0.62
DIST	0.95	0.90	0.92
TEMP	1	0.96	0.98
LOC	0.86	0.85	0.84
ORG	0.47	0.75	0.57
NUM	0.72	1	0.85
PER	0.82	0.67	0.73
OBJ	0.5	0.2	0.29
MISC	0.17	0.13	0.14

Table 8: Success rates using Naive Bayes Classifier

Classes	Precision	Recall	F–1 Score
MNY	0.79	0.69	0.73
DIST	1	0.95	0.98
TEMP	0.68	0.96	0.80
LOC	0.86	0.85	0.84
ORG	0.47	0.71	0.57
NUM	0.99	1	0.96
PER	0.83	0.89	0.86
OBJ	0.75	0.30	0.45
MISC	0.17	0.125	0.14

3.3 Comparative Analysis

In this subtask-1, total of 20 system runs is submitted by 7 teams and as an average 140 questions are successfully tagged by these teams with an average of 40 unsuccessful tags. For this task, *IINTU* team achieved 83.33333% as highest accuracy and our team *NLP–NITMZ* got the highest accuracy as 78.88889%. Among these 9 question classes, 'DIST' class has the highest precession value as 0.9903 and 'NUM' class got the highest recall value as 0.9961 and **temporal** class achieved 0.9612 as highest F–1 score.

4. CONCLUSIONS

We submitted 3 system runs and accuracy of our systems are 74.44%, 78.89%, and 74.44% respectively using three methods. For this subtask–1 of *MSIR16*, our system has given the best performance using Method–2 and we submitted the output of this method as run–3. For this run, two

Table 9: Success rates using Direct and Dependent rules

Classes	Precision	Recall	F–1 Score
MNY	0.91	0.63	0.74
DIST	0.95	0.90	0.93
TEMP	1	0.92	0.96
LOC	0.77	0.87	0.82
ORG	0.88	0.58	0.70
NUM	0.81	1	0.90
PER	0.83	0.89	0.86
OBJ	0.38	0.30	0.33
MISC	0.17	0.25	0.20

types of features are working together and we have given *direct* and *dependent* as the name of two sets. Between these two sets, *dependent* features are mainly worked for tagging the questions without ambiguity and we got the 7^{th} and 9^{th} rank with the system run 3 and 2.

5. ACKNOWLEDGMENTS

This work presented here under the research project Grant No. YSS/2015/000988 and supported by the Department of Science & Technology (DST) and Science and Engineering Research Board (SERB), Govt. of India. Authors have also acknowledged the Department of Computer Science & Engineering of National Institute of Technology Mizoram, India for proving infrastructural facilities.

6. REFERENCES

[1] S. Banerjee, K. Chakma, S. K. Naskar, A. Das, P. Rosso, S. Bandyopadhyay, and M. Choudhury. Overview of the Mixed Script Information Retrieval (MSIR) at FIRE. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[2] S. Banerjee, S. K. Naskar, P. Rosso, and S. Bandyopadhyay. The First Cross-Script Code-Mixed Question Answering Corpus. *Proceedings of the workshop on Modeling, Learning and Mining for Cross/Multilinguality (MultiLingMine 2016), co-located with The 38th European Conference on Information Retrieval (ECIR)*, 2016.

[3] B. Gambäck and A. Das. Comparing the level of code-switching in corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, May 2016.

[4] P. Gupta, K. Bali, R. E. Banchs, M. Choudhury, and P. Rosso. Query expansion for mixed-script information retrieval. In *The 37th Annual ACM SIGIR Conference, SIGIR-2014*, pages 677–686, Gold Coast, Australia, June 2014.

[5] X. Li and D. Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[6] P. Pakray and G. Majumder. Nlp–nitmz:part-of-speech tagging on italian social media text using hidden markov model. techreport, (Acceptd) In the SHARED TASK ON PoSTWITA – POS tagging for Italian Social Media Texts, EVALITA -2016.