

Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет «Высшая школа экономики»»
Факультет компьютерных наук
Образовательная программа Прикладная математика и информатика
Направление подготовки 01.03.02 Прикладная математика и информатика
бакалавриат

ОТЧЕТ

по преддипломной практике

Выполнил студент гр. 171
Биршерт Алексей Дмитриевич

Проверила:
Доцент,
Артемова Екатерина Леонидовна

23.04.2021

2021 год

Содержание

1	Введение	3
1.1	Цели и задачи практики	3
1.2	Постановка задачи	3
1.3	Актуальность темы	3
2	Обзор литературы	5
3	Описание методов	6
3.1	Общий вид атаки	6
3.2	Первый метод - word level attack	6
3.3	Второй метод - phrase level attack	7
3.4	Третий метод - slots chunk-level attack	7
3.5	Полученные результаты	7
4	Заключение	9

1 Введение

1.1 Цели и задачи практики

Цель данной учебной практики заключалась в подготовке исследования в рамках Выпускной квалификационной работы на тему "Атаки на мультязычные модели".

Для успешного прохождения практики были поставлены следующие задачи:

- 1 Обучение различных мультязычных языковых моделей на датасете ATIS - Seven languages [7].
- 2 Генерация дополнительных тестовых выборок с помощью различных методов адверсариальных атак.
- 3 Сравнение полученных результатов на всех выборках и анализ адверсариальных атак.

1.2 Постановка задачи

Основная задача практики заключается в анализе различных адверсариальных атак на мультязычные языковые модели. Модели должны быть обучены на датасете ATIS - Seven languages [7] для задачи одновременного выделения слотов и классификации интенгов пользователя.

Каждая из рассматриваемых адверсариальных атак состоит в различных пертурбациях тестовой выборки, а именно смещении языков внутри одного предложения. Результатом практики будет сравнение полученных результатов для каждой из атак для каждой из моделей.

Если у нас есть целевая модель \mathcal{M} , пример из тестовой выборки x с метками y , то наша цель найти такую пертурбацию x , которая максимизирует ошибку модели \mathcal{M} .

$$x' = \arg \max_{x_c \in X} \mathcal{L}(y, \mathcal{M}(x_c)),$$

где $x_c \in X$ это адверсариальная пертурбация x .

1.3 Актуальность темы

В последнее время создаётся всё больше мультязычных моделей и появляется всё больше исследований на тему межъязыкового обобщения. Новые методы и модели показывают впечатляющие результаты в переносе знаний и дообучении. Однако перенос с

одного языка на другой недостаточен для таких моделей для полноценного понимания мультязычных людей в мультязычных сообществах по всему миру. Во многих из этих сообществ смешение языков внутри одного предложения или фразы является повседневной практикой. Это называется код-свитчинг, феномен специфичный для мультязычных сред, возникающий как в обычных разговорах, так и в переписках и постах в интернете. Таким образом, для систем обработки естественного языка важно уметь работать и показывать хорошее качество на таких входных данных. Существуют вручную собранные и размеченные датасеты с код-свитчингом, которые позволяют оценить реальное качество моделей и дообучить их. Но сбор и разметка таких датасетов очень дорогие, так же возможное количество смешений различных языков является большой проблемой. Мы постулируем, что качество модели на искусственно сгенерированных с использованием адверсариальных атак тестовых данных может служить нижней оценкой на качество модели на реальных данных с код-свитчингом. Эти сгенерированные данные будут служить "самым плохим случаем что позволит думать, что в случае реальных данных качество модели будет выше.

2 Обзор литературы

Наше исследование опирается на несколько статей. Первая и основная статья [6] вышла полтора месяца назад и содержит в себе примерно такую же идею, что и в нашем исследовании. В статье описываются два варианта адверсариальных атак, оба из которых мы собираемся утилизировать и апробировать для нашей задачи. Первый метод заключается в переводе слов в предложении. Второй метод заключается в построении выравниваний между исходным предложением и его переводом и заменой слова или фразы на их отображение из предложения на другом языке. Во всех атаках замены слова/фразы выбираются с целью максимизации ошибки модели. Также в статье описывается метод обучения мультязычных моделей. Метод постулируется как способный помочь моделям показывать лучшее качество на код-свитчинге.

Вторая статья [5] также вышла полтора месяца назад и описывает еще один метод адверсариальной атаки, который мы собираемся использовать. Этот метод заключается в сегментации предложения и его переводов по меткам слотов и случайного перемешивания сегментированных частей между различными языками.

3 Описание методов

3.1 Общий вид атаки

Algorithm 1 Adversarial attack

Require: Пара пример-метка x, y ; целевая модель \mathcal{M} ; набор встраиваемых языков \mathbb{L}

Ensure: Адверсариальный пример x'

```
 $\mathcal{L}_x = \text{GetLoss}(\mathcal{M}, x, y)$ 
for  $i$  in permutation(len(x)) do
    Candidates, Losses = GetCandidates( $\mathcal{M}, x, y$ , token_id =  $i$ )
    if Candidates is not None and max(Losses) >  $\mathcal{L}_x$  then
         $\mathcal{L}_x = \max(\text{Losses})$ 
         $x[i] = \text{Candidates}[\text{argmax}(\text{Losses})]$ 
    end if
end for
return  $x$ 
```

3.2 Первый метод - word level attack

В качестве первой атаки была выбрана атака на уровне слов по аналогии с атакой PolyGloss из [6]. Для перевода используются словари из статьи [1].

Algorithm 2 Word-level attack

Require: Набор словарей с исходного на встраиваемые языки \mathbb{T}

```
function GETCANDIDATES( $\mathcal{M}, x, y$ , token_id)
    Candidates, Losses = [ ], [ ]
     $x_c = \text{copy}(x)$ 
    for language in  $\mathbb{L}$  do
        if  $x[\text{token\_id}]$  in  $\mathbb{T}[\text{language}]$  then
            token =  $\mathbb{T}[\text{language}][x[\text{token\_id}]]$ 
            Candidates.append(token)
             $x_c[\text{token\_id}] = \text{token}$ 
            Losses.append(GetLoss( $\mathcal{M}, x_c, y$ ))
        end if
    end for
return Candidates, Losses
end function
```

3.3 Второй метод - phrase level attack

В качестве второй атаки была выбрана атака на уровне фраз с использованием выравниваний по аналогии с атакой Bumblebee из [6]. Для построения выравниваний между предложениями используется метод, описанный в статье [4].

Algorithm 3 Phrase-level attack

Require: Выравнивание предложений с исходного на встраиваемые языки \mathbb{A}

```
function GETCANDIDATES( $\mathcal{M}, x, y, \text{token\_id}$ )
  Candidates, Losses = [ ], [ ]
   $x_c = \text{copy}(x)$ 
  for language in  $\mathbb{L}$  do
    if token_id in  $\mathbb{A}[\text{language}]$  then
      tokens =  $\mathbb{A}[\text{language}][\text{token\_id}]$ 
      Candidates.append(tokens)
       $x_c[\text{token\_id}] = \text{tokens}$ 
       $y_{\text{slots}}[\text{token\_id}] = \text{ExtendLabels}(y_{\text{slots}}[\text{token\_id}], \text{tokens})$ 
      Losses.append(GetLoss( $\mathcal{M}, x_c, y$ ))
    end if
  end for
  return Candidates, Losses
end function
```

3.4 Третий метод - slots chunk-level attack

В качестве третьего варианта атаки можно выбрать атаку по методу из статьи [5].

3.5 Полученные результаты

На данный момент обучено две мультязычные модели - XLM-Roberta [2] и M-BERT [3]. Модели обучены на обучающей выборке датасета ATIS - Seven languages [7]. На каждую из этих моделей проведено несколько адверсариальных атак с использованием первого и второго методов атак. Дальнейшая работа состоит в анализе полученных результатов атак.

	No attack	Word level [all]	Word level [de]	Word level [es]	Word level [fr]	Word level [ja]	Word level [pt]	Word level [zh_cn]
intent_acc	0.963	0.307	0.635	0.693	0.733	0.648	0.757	0.647
slot_precision	0.947	0.125	0.48	0.444	0.496	0.306	0.46	0.515
slot_recall	0.942	0.101	0.438	0.359	0.484	0.346	0.427	0.543
slot_f1	0.944	0.112	0.458	0.397	0.49	0.325	0.443	0.528
semantic_frame_acc	0.76	0.0	0.039	0.017	0.024	0.007	0.021	0.047
loss	0.477	11.537	4.791	4.823	4.401	5.76	3.941	5.075
time	1.793	349.559	63.845	65.729	63.149	69.834	62.836	66.602

Таблица 1: Результаты для модели XLM-R для атаки word-level attack

	No attack	Word level [ALL]	Word level [de]	Word level [es]	Word level [fr]	Word level [ja]	Word level [pt]	Word level [zh_cn]
intent_acc	0.964	0.255	0.684	0.731	0.728	0.633	0.75	0.652
slot_precision	0.943	0.13	0.406	0.416	0.479	0.374	0.437	0.551
slot_recall	0.939	0.105	0.388	0.369	0.501	0.351	0.414	0.574
slot_f1	0.941	0.116	0.397	0.391	0.49	0.362	0.425	0.563
sementic_frame_acc	0.766	0.0	0.013	0.016	0.028	0.013	0.015	0.069
loss	0.42	11.143	4.633	4.238	3.945	5.552	3.72	4.267
time	1.78	341.357	62.362	63.954	61.017	66.358	60.998	63.967

Таблица 2: Результаты для модели M-BERT для атаки word-level attack

	No attack	Alignments [ALL]	Alignments [de]	Alignments [es]	Alignments [fr]	Alignments [ja]	Alignments [pt]	Alignments [zh_cn]
intent_acc	0.963	0.821	0.918	0.885	0.896	0.887	0.876	0.884
slot_precision	0.947	0.379	0.739	0.711	0.667	0.374	0.718	0.496
slot_recall	0.942	0.483	0.756	0.696	0.698	0.498	0.736	0.673
slot_f1	0.944	0.425	0.747	0.703	0.683	0.427	0.727	0.571
sementic_frame_acc	0.76	0.086	0.382	0.281	0.3	0.067	0.354	0.169
loss	0.475	5.366	1.958	2.053	2.403	4.51	2.193	3.26
time	1.437	351.768	66.849	69.736	69.263	56.668	69.087	68.256

Таблица 3: Результаты для модели XLM-R для атаки phrase-level attack

	No attack	Alignments [ALL]	Alignments [de]	Alignments [es]	Alignments [fr]	Alignments [ja]	Alignments [pt]	Alignments [zh_cn]
intent_acc	0.964	0.832	0.922	0.906	0.909	0.898	0.904	0.892
slot_precision	0.943	0.365	0.716	0.704	0.659	0.368	0.707	0.482
slot_recall	0.939	0.447	0.731	0.7	0.682	0.466	0.72	0.66
slot_f1	0.941	0.402	0.723	0.702	0.67	0.411	0.713	0.557
sementic_frame_acc	0.766	0.063	0.365	0.265	0.283	0.058	0.363	0.159
loss	0.424	4.856	1.836	1.772	2.062	4.033	1.98	2.911
time	1.405	345.974	65.046	67.657	67.118	55.104	65.252	62.554

Таблица 4: Результаты для модели M-BERT для атаки phrase-level attack

4 Заключение

На данном этапе можно считать полностью выполненными задачи практики. Дальнейшая работа будет заключаться в расширении количества используемых методов атак, увеличении количества моделей и анализе получаемых результатов.

Список литературы

- [1] Yo Joong Choe, Kyubyong Park, and D. Kim. word2word: A collection of bilingual lexicons for 3, 564 language pairs. In *LREC*, 2020.
- [2] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, E. Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *ACL*, 2020.
- [3] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.
- [4] Zi-Yi Dou and Graham Neubig. Word alignment by fine-tuning embeddings on parallel corpora. In *EACL*, 2021.
- [5] Jitin Krishnan, Antonios Anastasopoulos, Hemant Purohit, and H. Rangwala. Multilingual code-switching for zero-shot cross-lingual intent prediction and slot filling. *ArXiv*, abs/2103.07792, 2021.
- [6] Samson Tan and Shafiq Joty. Code-mixing on sesame street: Dawn of the adversarial polyglots. *ArXiv*, abs/2103.09593, 2021.
- [7] Weijia Xu, Batool Haider, and Saab Mansour. End-to-end slot alignment and recognition for cross-lingual nlu. *ArXiv*, abs/2004.14353, 2020.