Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет
«Высшая школа экономики»

Факультет компьютерных наук
Основная образовательная программа
Прикладная математика и информатика

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Исследовательский проект на тему
"Атаки на мультиязычные модели"

Выполнил студент группы 171, 4 курса,
Биршерт Алексей Дмитриевич

Руководитель ВКР:
Доцент,
Департамент больших данных и информационного поиска
Артемова Екатерина Леонидовна

Москва 2021

Содержание

Аннотация

Какие-то слова в абстракте. Какие-то слова в абстракте.
Ссылка на гитхаб с проектом - https://github.com/birshert/attack-lang-models.

Ключевые слова—Ключевые слова

Some words in abstract. Some words in abstract.
Github project link - https://github.com/birshert/attack-lang-models.

Keywords—Keywords

1 Введение

2 Обзор литературы

2.1 Что-то первое

2.2 Что-то второе

2.3 Что-то третье

2.4 Что-то четвертое

3 Основная часть

3.1 Обучение моделей на датасете ATIS seven languages

В своей работе мы обучаем языковые модели решать задачу одновременной классификации интентов и разметки слотов в предложении. Эта задача заключается в определении желаемой цели запроса пользователя по предложению и классификации слов в предложении.

3.1.1 Датасет

В качестве датасета в своей работе мы выбрали датасет ATIS seven languages [6]. В этом датасете представлены семь языков из трёх языковых семей — Индо-Европейская (английский, немецкий, французский, испанский, португальский), Японо-рюкюская (японский) и Сино-тибетская (китайский). Датасет является параллельным корпусом для задачи классификации интентов и разметки слотов - в 2020 году он был переведён с английского языка на остальные шесть. В обучающей выборке содержится 4978 предложений для каждого языка, в тестовой 893 предложения для каждого языка.

Каждый объект в датасете состоит из предложения, меток слов и интента. Перед началом работы с датасетом мы произвели предварительную очистку — убрали из обучающей и тестовой выборок объекты, для которых на любом из семи языков количество слов и количество слотов не совпадали. Таким образом, в обучающей выборке осталось 4884 объекта для каждого языка, в тестовой выборке 755 объектов для каждого языка. Для составления списка используемых слотов и интентов использовалась обучающая выборка на английском языке. Мы использовали 121 различную метку слотов и 23 различных метки интентов. Список id используемых объектов, а также списки используемых слотов и интентов можно найти в приложении.

3.1.2 Архитектура модели

В своей работе мы решаем задачу одновременной классификации интентов и разметки слотов в предложении с помощью одной модели. Модель имеет два выхода, первый предсказывает интенты, второй предсказывает метки слов. В качестве рассматриваемых архитектур были выбраны модели m-BERT [2] и XLM-RoBERTa [1]. Обе эти модели являются одними из самых сильных мультиязычных моделей на текущий момент. Каждая из них предобучена на более чем ста языках.

3.1.3 Обучение

В своей работе мы будем сравнивать модели, обученные на всей обучающей выборке и только на части обучающей выборки на английском языке. Таким образом мы сможем проверить гипотезу о наличии кросс-язычных знаний у моделей. Тестовая выборка, которая будет нас интересовать в данном контексте состоит из всех семи языков, но мы оцениваем качество на каждом языке отдельно.

Каждая из моделей обучалась с одинаковыми гиперпараметрами - 10 эпох на обучающей выборке с длиной шага обучения 10^{-5} и размером батча в 64 объекта.

В своей работе мы будем использовать следующие метрики качества:

- Доля правильных ответов для интентов:

$$\textbf{Intent accuracy}(x, y) = \frac{1}{N}\sum_{i=1}^{N}[x_i = y_i] \tag{1}$$

- F1 мера для меток слотов:

$$\textbf{Slots F1 score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2}$$

- Доля предложений, в которых правильно классифицирован интент и

верно классифицированы все слоты:

$$\textbf{Semantic accuracy} = \#sentences[(I_{pred} = I_{true}) \wedge (S_{pred} = S_{true})] \quad (3)$$

3.2 Адверсариальные атаки

В своей работе мы предлагаем два варианта gray-box адверсариальных атак — во время выполнения атаки мы имеем доступ к ошибке модели. Мы стремимся создать атаку такого рода, чтобы результирующая адверсариальная пертурбация предложения была как можно ближе к реалистичным предложениям со смешением кодов. Для этого мы заменяем часть токенов в предложении на их эквиваленты из других языков. Оценка качества на таких адверсариальных атаках может выступать в роли оценки снизу на качество соответствующих моделей в аналогичных задачах при наличии реального смешения кодов во входных данных.

Так как большинство людей, которые могут использовать смешение кодов в своей речи билингвы, то в основном смешение кодов происходит между парой языков [5]. Таким образом, в своей работе мы предлагаем анализировать атаки состоящие во встраивании одного языка в другой.

3.2.1 Общий вид атаки

Общий принцип атаки одинаковый для обоих предлагаемых вариантов. Разница между методами заключается в способе генерации кандидатов на замену токену на $i-$ой позиции. В своей работе мы предлагаем следующий вид атаки — пусть мы имеем целевую модель, пару пример-метка и встраиваемый язык (1). Тогда мы перебираем токены в предложении в случайном порядке и стремимся заменить токен на его эквивалент из встраиваемого языка. Если это приведёт к увеличению ошибки модели, то мы заменяем токен на предложенного кандидата.

Algorithm 1 Адверсариальная атака, общая схема

Require: Пара пример-метка x, y; целевая модель $\mathcal{M}$; встраиваемый язык $\mathbb{L}$
Ensure: Адверсариальный пример x'

```
L_x = GetLoss(M, x, y)
for i in permutation(len(x)) do
    Candidates = GetCandidates(M, x, y, token_id = i)
    Losses = GetLoss(M, Candidates)
    if Candidates and max(Losses) > L_x then
        L_x = max(Losses)
        x, y = Candidates[argmax(Losses)]
    end if
end for
    return x
```

3.2.2 Word level атака

Первый предлагаемый нами вариант атаки заключается в генерации эквивалентов из других языков с помощью перевода токенов на соответствующие языки. Этот вариант является грубой оценкой снизу, так как он не учитывает контекста предложений и не учитывает многозначность слов.

Для перевода слов на другие языки мы используем модель машинного перевода M2M 100 от компании Facebook [4]. Она содержит 418 миллионов параметров.

Algorithm 2 Word-level атака

Require: Словарь переводов с исходного на встраиваемый язык $\mathbb{T}$

```
function GetCandidates(M, x, y, token_id)
    if x[token_id] in T[L] then
        tokens = T[L][x[token_id]]
        x[token_id] = tokens
        y[token_id] = ExtendSlotLabels(y[token_id], len(tokens))
    end if
    return x, y
end function
```

3.2.3 Phrase-level атака

Второй предлагаемый нами вариант атаки заключается в генерации эквивалентов из других языков с помощью построения выравниваний между

предложениями на разных языков. Кандидаты для каждого токена определяются как токены из предложения на встраиваемом языке, в которые был выровнен токен.

Для построения выравниваний мы используем модель awesome-align на основе m-BERT [3].

Algorithm 3 Word-level атака

Require: Выравнивание предложения на исходном языке к предложению на целевом языке $\mathbb{A}$

```
function GetCandidates(M, x, y, token_id)
    if x[token_id] in A[L] then
        tokens = A[L][x[token_id]]
        x[token_id] = tokens
        y[token_id] = ExtendSlotLabels(y[token_id], len(tokens))
    end if
    return x, y
end function
```

3.3 Метод защиты от адверсариальных атак

В своей работе мы предлагаем метод защиты от предложенных выше адверсариальных атак. Гипотеза заключается в том, что данный метод позволит увеличить качество не только на адверсариальных пертурбациях, но и на реальных данных со смешением кодов.

3.3.1 Метод адверсариального предобучения

3.4 Результаты

3.4.1 Кросс-язычные знания в моделях

	xlm-r	xlm-r en	xlm-r adv	xlm-r en + adv
Intent accuracy	0.980	0.902	0.981	0.928
Slot F1 score	0.944	0.870	0.947	0.888
Semantic accuracy	0.826	0.559	0.833	0.613
Loss	0.317	0.729	0.320	0.621

Таблица 1: Сравнение моделей XLM-R между собой на тестовой выборке (английский язык)

	m-bert	m-bert en	m-bert adv	m-bert en + adv
Intent accuracy	0.979	0.952	0.975	0.959
Slot F1 score	0.947	0.899	0.950	0.900
Semantic accuracy	0.854	0.672	0.861	0.674
Loss	0.353	0.584	0.326	0.567

Таблица 2: Сравнение моделей M-BERT между собой на тестовой выборке (английский язык)

	xlm-r	xlm-r en	xlm-r adv	xlm-r en + adv
Intent accuracy	0.969 ± 0.004	0.840 ± 0.044	0.970 ± 0.004	0.860 ± 0.043
Slot F1 score	0.928 ± 0.011	0.669 ± 0.063	0.930 ± 0.013	0.675 ± 0.113
Semantic accuracy	0.775 ± 0.044	0.181 ± 0.107	0.781 ± 0.048	0.245 ± 0.167
Loss	0.399 ± 0.055	1.498 ± 0.368	0.409 ± 0.063	1.453 ± 0.525

Таблица 3: Сравнение моделей XLM-R между собой на тестовой выборке (все языки кроме английского)

	m-bert	m-bert en	m-bert adv	m-bert en + adv
Intent accuracy	0.964 ± 0.008	0.828 ± 0.043	0.967 ± 0.006	0.837 ± 0.072
Slot F1 score	0.927 ± 0.020	0.616 ± 0.093	0.929 ± 0.015	0.576 ± 0.101
Semantic accuracy	0.776 ± 0.064	0.204 ± 0.103	0.779 ± 0.055	0.219 ± 0.117
Loss	0.425 ± 0.093	1.584 ± 0.348	0.382 ± 0.057	1.794 ± 0.768

Таблица 4: Сравнение моделей M-BERT между собой на тестовой выборке (все языки кроме английского)

3.4.2 Качество моделей после адверсариальных атак

	xlm-r	xlm-r en	xlm-r adv	xlm-r en + adv
Intent accuracy	0.876 ± 0.034	0.721 ± 0.086	0.888 ± 0.033	0.769 ± 0.079
Slot F1 score	0.642 ± 0.083	0.550 ± 0.068	0.640 ± 0.088	0.554 ± 0.076
Semantic accuracy	0.177 ± 0.101	0.065 ± 0.063	0.174 ± 0.102	0.101 ± 0.070
Loss	2.662 ± 0.737	3.234 ± 0.807	2.553 ± 0.669	2.905 ± 0.562

Таблица 5: Сравнение моделей XLM-R после word-level атаки

	m-bert	m-bert en	m-bert adv	m-bert en + adv
Intent accuracy	0.863 ± 0.025	0.771 ± 0.028	0.893 ± 0.017	0.819 ± 0.047
Slot F1 score	0.553 ± 0.098	0.447 ± 0.084	0.581 ± 0.085	0.455 ± 0.069
Semantic accuracy	0.117 ± 0.075	0.055 ± 0.051	0.155 ± 0.085	0.081 ± 0.054
Loss	3.169 ± 0.689	3.338 ± 0.667	2.860 ± 0.687	2.959 ± 0.574

Таблица 6: Сравнение моделей M-BERT после word-level атаки

	xlm-r	xlm-r en	xlm-r adv	xlm-r en + adv
Intent accuracy	0.949 ± 0.011	0.727 ± 0.131	0.952 ± 0.011	0.827 ± 0.035
Slot F1 score	0.708 ± 0.139	0.584 ± 0.111	0.716 ± 0.147	0.621 ± 0.146
Semantic accuracy	0.368 ± 0.161	0.106 ± 0.071	0.392 ± 0.156	0.214 ± 0.133
Loss	2.032 ± 1.156	2.860 ± 0.839	2.032 ± 1.233	2.113 ± 0.637

Таблица 7: Сравнение моделей XLM-R после phrase-level атаки

	m-bert	m-bert en	m-bert adv	m-bert en + adv
Intent accuracy	0.941 ± 0.006	0.829 ± 0.018	0.951 ± 0.005	0.847 ± 0.054
Slot F1 score	0.700 ± 0.127	0.538 ± 0.097	0.725 ± 0.142	0.578 ± 0.132
Semantic accuracy	0.345 ± 0.128	0.110 ± 0.055	0.424 ± 0.159	0.214 ± 0.116
Loss	2.131 ± 1.138	2.463 ± 0.585	1.970 ± 1.196	2.159 ± 0.755

Таблица 8: Сравнение моделей M-BERT после phrase-level атаки

3.4.3 Влияние метода адверсариального предобучения

4 Заключение

AAAAAAAAAAAAAAAAAA FUCK ME

Список литературы

[1] Alexis Conneau и др. «Unsupervised Cross-lingual Representation Learning at Scale». В: *ACL*. 2020.

[2] Jacob Devlin и др. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». В: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, июнь 2019, с. 4171—4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423.

[3] Zi-Yi Dou и Graham Neubig. «Word Alignment by Fine-tuning Embeddings on Parallel Corpora». В: *EACL*. 2021.

[4] Angela Fan и др. «Beyond English-Centric Multilingual Machine Translation». В: *ArXiv* abs/2010.11125 (2020).

[5] Shana Poplack, DAVID SANKOFF и CHRISTOPHER MILLER. «The social correlates and linguistic processes of lexical borrowing and assimilation». В: *Linguistics* 26 (янв. 1988), с. 47—104. DOI: 10.1515/ling.1988.26.1.47.

[6] Weijia Xu, Batool Haider и Saab Mansour. «End-to-End Slot Alignment and Recognition for Cross-Lingual NLU». В: *ArXiv* abs/2004.14353 (2020).

Приложения

Приложение А. Алгоритм замены слотов в атаке

Algorithm 4 Алгоритм замены слотов в атаке

```
function ExtendSlotLabels(slot_label, num_tokens)
    slot_labels = [slot_label]
    if num_tokens > 1 then
        if slot_label.startswith('B') then
            slot_labels += ['I' + slot_label[1:]] · (num_tokens - 1)
        else
            slot_labels ·= num_tokens
        end if
    end if
    return slot_labels
end function
```