

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 мультязычные 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

<b>1</b>	<b>Введение</b>	<b>4</b>
<b>2</b>	<b>Обзор литературы</b>	<b>5</b>
2.1	Мультиязычные модели . . . . .	5
2.2	Классификация интенгов и заполнение слотов . . . . .	5
2.3	Адверсариальные атаки на мультиязычные модели . . . . .	5
2.4	Машинный перевод и выравнивание слов . . . . .	5
<b>3</b>	<b>Основная часть</b>	<b>6</b>
3.1	Обучение моделей на датасете MultiAtis++ . . . . .	6
3.1.1	Датасет . . . . .	6
3.1.2	Архитектура модели . . . . .	7
3.1.3	Обучение . . . . .	7
3.2	Адверсариальные атаки . . . . .	8
3.2.1	Общий вид атаки . . . . .	9
3.2.2	Word level атака . . . . .	9
3.2.3	Phrase-level атака . . . . .	10
3.3	Метод адверсариального предобучения для защиты от адвер- сариальных атак . . . . .	10
3.3.1	Генерация адверсариальной выборки . . . . .	11
3.3.2	Дообучение тела модели . . . . .	12
3.3.3	Загрузка дообученного тела модели . . . . .	12
3.4	Результаты . . . . .	13
3.4.1	Кросс-язычные знания в моделях . . . . .	13
3.4.2	Качество моделей после адверсариальных атак . . . . .	14
3.4.3	Влияние метода адверсариального предобучения . . . . .	17
<b>4</b>	<b>Заключение</b>	<b>18</b>
	<b>Список литературы</b>	<b>19</b>

<b>Приложения</b>	<b>20</b>
Приложение А. Алгоритм замены слотов в атаке . . . . .	20

## 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. Some words in abstract. Some  
words in abstract. Some words in abstract. Some words in abstract. Some words in  
abstract. Some words in abstract. Some words in abstract. Some words in abstract.  
Some words in abstract. Some words in abstract. Some words in abstract. Some  
words in abstract. Some words in abstract. Some words in abstract. Some words in  
abstract. Some words in abstract. Some words in abstract. Some words in abstract.  
Some words in abstract. Some words in abstract. Some words in abstract. Some  
words in abstract. Some words in abstract. Some words in abstract. Some words in  
abstract. Some words in abstract. Some words in abstract. Some words in abstract.  
Some words in abstract. Some words in abstract. Some words in abstract. Some  
words in abstract. Some words in abstract. Some words in abstract. Some words  
in abstract.

Github project link - <https://github.com/birshert/attack-lang-models>.

**Keywords**—Keywords

# 1 Введение

Последние несколько лет стали прорывными в области мультязычных моделей и их обобщающей способности для других языков [6, 10]. Огромные мультязычные модели выучивают универсальные языковые представления, что помогает им демонстрировать удивительные способности к переносу знаний с одного языка на другой. Простое дообучение предобученных моделей для какой-либо задачи на языке с большим количеством данных позволяет достичь хорошего качества на других языках.

Однако простой перенос между языками недостаточен для систем обработки естественного языка для понимания мультязычных пользователей. Во многих сообществах в мире достаточно часто явление смешения кодов. Смешение кодов — это процесс, когда человек спонтанно смешивает различные языки внутри одного предложения или фразы. Такой феномен может проявляться как в письменной, так и в устной речи. Таким образом, важно сделать языковую модель устойчивой к смешению языков, чтобы модель адекватно работала со входными данными.

Несмотря на то, что реальные данные со смешением языков очень важны для оценки способности языковых моделей работы со смешением кодов, такие данные очень тяжело собирать и размечать в большом количестве.

В своей работе мы предполагаем, что качество моделей на адверсариальных атаках может служить нижней оценкой на реальное качество модели. Если языковая модель успешно справляется с адверсариальными пертурбациями со смешением кодов, то и в реальной жизни она будет успешно обрабатывать данные от мультязычных пользователей. Таким образом, мы в своей работе:

- Предлагаем две адверсариальные атаки по методу серого ящика — во время атаки мы имеем доступ к ошибке модели на заданных данных. Мы проводим атаки на мультязычные модели для задачи одновременного детектирования намерений пользователя и заполнения слотов для

диалоговых помощников, направленных на выполнение конкретной задачи. Насколько нам известно, это одни из первых мультязычных адверсариальных атак на данную задачу.

- Предлагаем метод адверсариального предобучения и показываем, что он увеличивает качество моделей на наших атаках.
- Дополнительно исследуем перенос знаний для задачи одновременной классификации интенгов и разметки слотов в предложении.

Все свои эксперименты мы будем проводить с современными мультязычными моделями - m-BERT [2] и XLM-RoBERTa [1]. В качестве датасета мы будем использовать корпус MultiAtis++ [11].

## 2 Обзор литературы

### 2.1 Мультязычные модели

BERT - [2, 6, 10] , XLM-RoBERTa - [1],

### 2.2 Классификация интенгов и заполнение слотов

[9]

### 2.3 Адверсариальные атаки на мультязычные модели

[8, 5]

### 2.4 Машинный перевод и выравнивание слов

Перевод - [4].

Выравнивание - [3].

## 3 Основная часть

### 3.1 Обучение моделей на датасете MultiAtis++

В своей работе мы обучаем языковые модели решать задачу одновременного детектирования намерений пользователя и заполнения слотов для диалоговых помощников, направленных на выполнение конкретной задачи. Эта задача заключается в классификации предложений и всех слов в предложении.

#### 3.1.1 Датасет

В качестве датасета в своей работе мы выбрали датасет MultiAtis++ [11]. 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 интенгов 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 в BIO формате и интенга. 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 интенгов использовалась обучающая выборка на английском языке. Мы использовали 121 различную метку слотов и 23 различных метки интенгов. Список id используемых объектов, а также списки используемых слотов и интенгов можно найти в приложении.

### 3.1.2 Архитектура модели

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

Обозначим количество блоков Трансформера за  $L$ , размер скрытых пред-ставлений за  $H$  и количество голов с внутренним вниманием за  $A$ . Тогда в используемой нами модели m-BERT  $L = 12$ ,  $H = 768$ ,  $A = 12$ , а суммарное количество параметров 110 миллионов. В используемой нами модели XLM-RoBERTa  $L = 12$ ,  $H = 768$ ,  $A = 12$ , а суммарное количество параметров 270 миллионов.

### 3.1.3 Обучение

В своей работе мы будем сравнивать модели, обученные на всей обучаю-щей выборке и только на части обучающей выборки на английском языке. Таким образом мы сможем проверить гипотезу о наличии кросс-язычных зна-ний 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 качестве функции ошибки использовалась кросс-энтропия:

$$L = -\frac{1}{n} \sum_{i=1}^n [y \log(\hat{y})] \quad (1)$$

В своей работе мы будем использовать следующие метрики качества:



- Доля предложений, в которых правильно классифицирован интен-  
т:

$$\text{Intent accuracy} = \# \text{sentences} [(I_{pred} = I_{true})] \quad (2)$$

- F1 мера для меток слотов (используется микро-усреднение по всем клас-  
сам):

$$\text{Slots F1 score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

- Доля предложений, в которых правильно классифицирован интен-  
т и  
верно классифицированы все слоты:

$$\text{Semantic accuracy} = \# \text{sentences} [(I_{pred} = I_{true}) \wedge (S_{pred} = S_{true})] \quad (4)$$

## 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 [7]. 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 токenu 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** Общая схема адверсариальной атаки

---

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

**Ensure:** Адверсариальный пример  $x'$

```
 $\mathcal{L}_x = \text{GetLoss}(\mathcal{M}, x, y)$ 
for  $i$  in  $\text{permutation}(\text{len}(x))$  do
    Candidates =  $\text{GetCandidates}(\mathcal{M}, x, y, \text{token\_id} = i)$ 
    Losses =  $\text{GetLoss}(\mathcal{M}, \text{Candidates})$ 
    if Candidates and  $\max(\text{Losses}) > \mathcal{L}_x$  then
         $\mathcal{L}_x = \max(\text{Losses})$ 
         $x, y = \text{Candidates}[\text{argmax}(\text{Losses})]$ 
    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( $\mathcal{M}$ ,  $x$ ,  $y$ , token_id)
  if  $x[\text{token\_id}]$  in  $\mathbb{T}[\mathbb{L}]$  then
    tokens =  $\mathbb{T}[\mathbb{L}][x[\text{token\_id}]]$ 
     $x[\text{token\_id}]$  = tokens
     $y[\text{token\_id}]$  = ExtendSlotLabels( $y[\text{token\_id}]$ , 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 языках. 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( $\mathcal{M}$ ,  $x$ ,  $y$ , token_id)
  if  $x[\text{token\_id}]$  in  $\mathbb{A}[\mathbb{L}]$  then
    tokens =  $\mathbb{A}[\mathbb{L}][x[\text{token\_id}]]$ 
     $x[\text{token\_id}]$  = tokens
     $y[\text{token\_id}]$  = ExtendSlotLabels( $y[\text{token\_id}]$ , len(tokens))
  end if
  return  $x$ ,  $y$ 
end function
```

---

## 3.3 Метод адверсариального предобучения для защиты от адверсариальных атак

В своей работе мы предлагаем метод защиты от предложенных выше адверсариальных атак. Гипотеза заключается в том, что данный метод позво-

лит увеличить качество не только на адверсариальных пертурбациях, но и на реальных данных со смещением кодов.

Предлагаемый нами метод адверсариального предобучения состоит из нескольких шагов:

- 1 Генерация выборки для задачи маскированного моделирования языка.
- 2 Дообучение тела мультязычной модели на сгенерированной выборке в режиме предсказания маскированных токенов.
- 3 Загрузка дообученного тела модели перед началом обучения для задачи одновременного заполнения слотов и классификации интенгов.

---

**Algorithm 4** Генерация адверсариальной выборки

---

**Require:** Обучающая выборка датасета  $X$ , набор встраиваемых языков  $\mathbb{L}_1, \dots, \mathbb{L}_n$

**Ensure:** Адверсариальная выборка  $X'$

$X' = []$

**for**  $\mathbb{L}$  in  $\mathbb{L}_1, \dots, \mathbb{L}_n$  **do**

**for**  $x$  in  $X$  **do**

**for**  $i$  in permutation(len( $x$ )) **do**

            Candidates = GetCandidates( $\mathcal{M}$ ,  $x$ ,  $y$ , token\_id =  $i$ )

**if** Candidates and  $\mathcal{U}(0, 1) > 0.5$  **then**

$x, \_ = \text{random.choice}(\text{Candidates})$

**end if**

**end for**

$X'.\text{append}(x)$

**end for**

**end for**

**return**  $X'$

---

### 3.3.1 Генерация адверсариальной выборки

Для генерации выборки используется адаптация алгоритма phrase-level адверсариальной атаки. Разница заключается в том, что токены заменяются на их эквиваленты с некоторой вероятностью. Таким образом, для генерации выборки не требуется обученная модель.

Выборка является конкатенацией сгенерированных выборок для всех шести языков кроме английского представленных в датасете. Каждая из подвыборок генерируется встраиванием целевого языка в обучающую выборку датасета MultiAtis++ на английском языке (4). Псевдокод функции GetCandidates представлен в секции про атаки (3).

После генерации у нас получается 6 подвыборок по 4884 предложения в каждой. Итоговая выборка состоит из 29304 предложений, мы делим эту выборку в отношении 9 к 1 на обучающую и тестовую.

### **3.3.2 Дообучение тела модели**

После генерации адверсариальной выборки мы дообучаем предобученную мультязычную модель на этой выборке. Модель обучается в режиме задачи маскированного моделирования языка.

Для обучения модели для такой задачи мы маскируем 15% токенов и предсказываем их с помощью модели. 80% маскированных токенов заменяются на токен маски, 10% заменяются на случайные слова из словаря, остальные 10% остаются неизменными. Мы дообучаем обе мультязычные модели m-BERT и XLM-RoBERTa с одинаковыми гиперпараметрами - 10 эпох с размером батча 64 и длиной шага  $10^{-5}$ . После дообучения мы сохраняем тело модели для дальнейшего использования.

### **3.3.3 Загрузка дообученного тела модели**

Перед обучением мультязычной модели для задачи одновременного заполнения слотов и классификации интенгов мы загружаем дообученное тело модели.

## 3.4 Результаты

### 3.4.1 Кросс-язычные знания в моделях

	en	de	es	fr	ja	pt	zh
<b>xlm-r</b>	0.980	0.976	0.967	0.971	0.970	0.967	0.964
<b>m-bert</b>	0.979	0.976	0.958	0.968	0.955	0.968	0.956
<b>xlm-r en</b>	0.902	0.875	0.878	0.879	0.785	0.775	0.848
<b>m-bert en</b>	0.952	0.820	0.870	0.875	0.747	0.838	0.816
<b>xlm-r adv</b>	0.981	0.974	0.964	0.976	0.972	0.967	0.967
<b>m-bert adv</b>	0.975	0.976	0.964	0.972	0.960	0.970	0.962
<b>xlm-r en + adv</b>	0.928	0.890	0.913	0.872	0.789	0.881	0.816
<b>m-bert en + adv</b>	0.959	0.848	0.901	0.893	0.719	0.901	0.759

Таблица 1: Сравнение моделей между собой на тестовой выборке датасета MultiAtis++ по метрике **Intent accuracy**

	en	de	es	fr	ja	pt	zh
<b>xlm-r</b>	0.944	0.939	0.908	0.924	0.929	0.924	0.942
<b>m-bert</b>	0.947	0.945	0.885	0.926	0.935	0.924	0.945
<b>xlm-r en</b>	0.870	0.669	0.751	0.612	0.573	0.673	0.738
<b>m-bert en</b>	0.899	0.558	0.783	0.534	0.622	0.518	0.680
<b>xlm-r adv</b>	0.947	0.940	0.906	0.929	0.928	0.929	0.946
<b>m-bert adv</b>	0.950	0.942	0.900	0.928	0.935	0.920	0.946
<b>xlm-r en + adv</b>	0.888	0.729	0.788	0.623	0.447	0.743	0.718
<b>m-bert en + adv</b>	0.900	0.566	0.759	0.557	0.416	0.554	0.604

Таблица 2: Сравнение моделей между собой на тестовой выборке датасета MultiAtis++ по метрике **Slots F1 score**

	en	de	es	fr	ja	pt	zh
<b>xlm-r</b>	0.826	0.826	0.697	0.807	0.739	0.800	0.779
<b>m-bert</b>	0.854	0.854	0.653	0.804	0.740	0.808	0.796
<b>xlm-r en</b>	0.559	0.250	0.347	0.151	0.000	0.204	0.132
<b>m-bert en</b>	0.672	0.188	0.401	0.195	0.052	0.177	0.213
<b>xlm-r adv</b>	0.833	0.832	0.693	0.813	0.746	0.809	0.792
<b>m-bert adv</b>	0.861	0.854	0.682	0.805	0.738	0.799	0.797
<b>xlm-r en + adv</b>	0.613	0.397	0.404	0.109	0.005	0.419	0.136
<b>m-bert en + adv</b>	0.674	0.266	0.366	0.265	0.004	0.278	0.136

Таблица 3: Сравнение моделей между собой на тестовой выборке датасета MultiAtis++ по метрике **Semantic accuracy**

### 3.4.2 Качество моделей после адверсариальных атак

	de	es	fr	ja	pt	zh
<b>xlm-r</b>	0.931	0.877	0.849	0.825	0.901	0.872
<b>m-bert</b>	0.893	0.891	0.872	0.820	0.853	0.852
<b>xlm-r en</b>	0.809	0.783	0.774	0.677	0.554	0.728
<b>m-bert en</b>	0.811	0.760	0.793	0.723	0.760	0.777
<b>xlm-r adv</b>	0.935	0.885	0.895	0.838	0.918	0.857
<b>m-bert adv</b>	0.923	0.894	0.890	0.866	0.899	0.885
<b>xlm-r en + adv</b>	0.842	0.817	0.812	0.613	0.811	0.721
<b>m-bert en + adv</b>	0.865	0.828	0.854	0.760	0.856	0.750

Таблица 4: Сравнение моделей между собой после word-level атаки на тестовую выборку датасета MultiAtis++ по метрике **Intent accuracy**

	de	es	fr	ja	pt	zh
<b>xlm-r</b>	0.767	0.589	0.603	0.552	0.598	0.747
<b>m-bert</b>	0.685	0.517	0.510	0.428	0.494	0.684
<b>xlm-r en</b>	0.642	0.467	0.499	0.508	0.543	0.641
<b>m-bert en</b>	0.539	0.385	0.419	0.362	0.391	0.585
<b>xlm-r adv</b>	0.768	0.609	0.591	0.519	0.608	0.744
<b>m-bert adv</b>	0.704	0.532	0.535	0.470	0.563	0.685
<b>xlm-r en + adv</b>	0.648	0.509	0.508	0.455	0.542	0.663
<b>m-bert en + adv</b>	0.531	0.405	0.371	0.441	0.417	0.563

Таблица 5: Сравнение моделей между собой после word-level атаки на тестовую выборку датасета MultiAtis++ по метрике **Slots F1 score**

	de	es	fr	ja	pt	zh
<b>xlm-r</b>	0.343	0.109	0.086	0.170	0.077	0.278
<b>m-bert</b>	0.228	0.083	0.058	0.081	0.038	0.213
<b>xlm-r en</b>	0.201	0.057	0.056	0.013	0.023	0.042
<b>m-bert en</b>	0.136	0.029	0.032	0.005	0.011	0.115
<b>xlm-r adv</b>	0.347	0.134	0.093	0.099	0.089	0.283
<b>m-bert adv</b>	0.302	0.114	0.087	0.121	0.070	0.237
<b>xlm-r en + adv</b>	0.236	0.094	0.065	0.012	0.073	0.128
<b>m-bert en + adv</b>	0.187	0.073	0.036	0.044	0.038	0.110

Таблица 6: Сравнение моделей между собой после word-level атаки на тестовую выборку датасета MultiAtis++ по метрике **Semantic accuracy**



	de	es	fr	ja	pt	zh
<b>xlm-r</b>	0.954	0.946	0.928	0.952	0.964	0.950
<b>m-bert</b>	0.948	0.935	0.939	0.951	0.940	0.934
<b>xlm-r en</b>	0.808	0.836	0.740	0.750	0.442	0.784
<b>m-bert en</b>	0.809	0.833	0.834	0.805	0.861	0.829
<b>xlm-r adv</b>	0.959	0.958	0.930	0.960	0.958	0.947
<b>m-bert adv</b>	0.956	0.950	0.944	0.958	0.955	0.946
<b>xlm-r en + adv</b>	0.870	0.857	0.811	0.795	0.853	0.776
<b>m-bert en + adv</b>	0.846	0.890	0.893	0.766	0.901	0.784

Таблица 7: Сравнение моделей между собой после phrase-level атаки на тестовую выборку датасета MultiAtis++ по метрике **Intent accuracy**

	de	es	fr	ja	pt	zh
<b>xlm-r</b>	0.802	0.829	0.751	0.444	0.813	0.609
<b>m-bert</b>	0.784	0.804	0.758	0.450	0.783	0.619
<b>xlm-r en</b>	0.627	0.704	0.569	0.365	0.680	0.561
<b>m-bert en</b>	0.539	0.699	0.531	0.366	0.530	0.563
<b>xlm-r adv</b>	0.809	0.847	0.771	0.432	0.820	0.617
<b>m-bert adv</b>	0.807	0.848	0.790	0.446	0.822	0.635
<b>xlm-r en + adv</b>	0.683	0.776	0.648	0.326	0.721	0.570
<b>m-bert en + adv</b>	0.613	0.754	0.621	0.324	0.631	0.523

Таблица 8: Сравнение моделей между собой после phrase-level атаки на тестовую выборку датасета MultiAtis++ по метрике **Slots F1 score**

	de	es	fr	ja	pt	zh
<b>xlm-r</b>	0.511	0.511	0.336	0.115	0.522	0.209
<b>m-bert</b>	0.487	0.438	0.344	0.114	0.433	0.256
<b>xlm-r en</b>	0.163	0.229	0.099	0.013	0.070	0.064
<b>m-bert en</b>	0.122	0.219	0.085	0.041	0.087	0.106
<b>xlm-r adv</b>	0.538	0.536	0.347	0.140	0.536	0.256
<b>m-bert adv</b>	0.544	0.551	0.466	0.136	0.559	0.286
<b>xlm-r en + adv</b>	0.303	0.375	0.196	0.008	0.322	0.079
<b>m-bert en + adv</b>	0.254	0.358	0.241	0.019	0.306	0.109

Таблица 9: Сравнение моделей между собой после phrase-level атаки на тестовую выборку датасета MultiAtis++ по метрике **Semantic accuracy**

Word-level атака значительно уменьшила качество для обеих моделей, причем M-BERT справился с этой атакой хуже.

### 3.4.3 Влияние метода адверсариального предобучения

## 4 Заключение

## 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)*. 2019, с. 4171—4186.
- [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] Jitin Krishnan и др. «Multilingual Code-Switching for Zero-Shot Cross-Lingual Intent Prediction and Slot Filling». В: *ArXiv abs/2103.07792* (2021).
- [6] Chi-Liang Liu и др. «What makes multilingual BERT multilingual?» В: *ArXiv abs/2010.10938* (2020).
- [7] Shana Poplack, DAVID SANKOFF и CHRISTOPHER MILLER. «The social correlates and linguistic processes of lexical borrowing and assimilation». В: *Linguistics* 26 (1988), с. 47—104.
- [8] Samson Tan и Shafiq Joty. «Code-Mixing on Sesame Street: Dawn of the Adversarial Polyglots». В: *ArXiv abs/2103.09593* (2021).
- [9] H. Weld и др. «A survey of joint intent detection and slot-filling models in natural language understanding». В: *ArXiv abs/2101.08091* (2021).
- [10] Shijie Wu и Mark Dredze. «Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT». В: *EMNLP/IJCNLP*. 2019.
- [11] 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 5** 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
```

---