МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М. В. ЛОМОНОСОВА
Факультет вычислительной математики и кибернетики
Кафедра алгоритмических языков

Куклин Михаил Игоревич

Использование языковой модели в задаче извлечения смысла слова

Курсовая работа

Научный руководитель:
Полякова И. Н.

Москва 2023

Содержание

1 Введение

Задача извлечения смысла слова (*Word Sense Induction*, WSI) имеет большую важность в области компьютерной лингвистики и обработки естественного языка. Она позволяет автоматически определять значения слова в зависимости от контекста, в котором оно употребляется.

Одной из основных проблем в обработке естественного языка является полисемия. Полисемия - это наличие нескольких значений у одного и того же слова. Например, слово "банк"может иметь значения как "финансовое учреждение так и "наклонная поверхность". Если не учитывать контекст, в котором это слово используется, может возникнуть неоднозначность его значения, что может привести к неправильному пониманию текста или вовсе искажению его смысла.

Автоматическое определение значений слова помогает устранить эту проблему и повысить точность анализа текстов. С помощью извлечения смысла слова можно определить, какое именно значение слова необходимо использовать в данном контексте, что позволяет улучшить качество анализа текста.

Одним из применений WSI является машинный перевод. Переводчики, как правило, используют контекст для определения значения слова, поэтому автоматическое определение значений слова может помочь повысить точность машинного перевода.

Извлечение смысла слова также является важным инструментом для информационного поиска. При поиске информации по ключевым словам, используемым в запросе, система может учитывать контекст и определять наиболее подходящее значение слова для более точного поиска.

Кроме того, WSI может помочь в анализе тональности текста. Значение слова может измениться в зависимости от того, в каком контексте оно используется, и это может влиять на тональность текста. Например, слово "хороший"может использоваться как с положительной, так и с отрицательной окраской в зависимости от контекста. Автоматическое определение значений слова позволяет точнее анализировать тональность текста.

WSI также может быть полезен в автоматическом реферировании. Автоматическое определение значений слова позволяет создать более точное и краткое изложение текста.

Таким образом, задача извлечения смысла слова имеет большое значение для обработки естественного языка и может быть применена в различных областях, где необходима точность анализа текста и определение значения слова в зависимости от контекста.

2 Цель работы

Целью данной работы является разработка и реализация программы, которая осуществляет кластеризацию текстов по смыслу целевого слова. К подзадачам данной работы относятся:

- Анализ существующих методов и подходов к решению задачи извлечения смысла слова.
- Програмная реализация модели и подбор оптимальных гиперпараметров.
- Сравнение полученных результатов с существующими решениями

3 Обзор существующих методов

Поставленную задачу можно разбить на два этапа: получение векторного представления примеров упротребления неоднозначных слов и их непосредственная кластеризация. Перед тем как перейти к решению задачи извлечения смысла слова, рассмотрим существующие методы векторизации и кластеризации.

3.1 Word2Vec

Перейдем к рассмотрению соревнования RUSSE'2018: a Shared Task on Word Sense Induction for the Russian Language [1], в котором участникам предстояла задача исзвлечения смысла многозначных слов русского языка. В качестве данных участникам были выданы 3 различных датасета. После завершения соревнования для каждого из датасетов были объявлены победители, а также их модели, которые они использовали для решения поставленной задачи. Рассмотрим 3 наилучших решения:

- Команда «jamstic» использовала CBOW [2], предобученный на Национальном корпусе русского языка, для получения неконтекстуализированного эмбеддинга целевого слова. Смысл слова извлекался с помощью метода ближайшего соседа. Сначала искалось первое слово, эмбеддинг которого больше всего похож на целевое слово. Затем бралась разность между эмбеддингами целевого слова и первого слова. Второе слово искалось аналогично первому, только вместо эмбеддинга целевого слова использовалось полученная разность. Получившиеся 2 слова и определяли смысл целевого слова. Для этого считалось среднее значение эмбеддингов контекста, после чего считалось косинусная близость до этих слов.

- Комнда «Pavel» использовала CBOW, предобученный на lib.rus.ec, после этого считалось средневзвешенное контекта, где вес слова определялся как $tfidf^{1.5}chisq^{0.5}$. [3][4] Затем использовалась агломеративная кластеризация[5].

- Команда «akutuzov» использовала предобученный skipgram[2], в котором считалось средневзвешенное эмбеддингов контекста, в качестве кластери-

зации был выбран алгоритм affinity propagation[6].

3.2 Лексические подстановки

Другим подходом к векторизации являются лексические подстановки – слова, которые могут стоять на месте целевого слова. В статье[7] описан следующий алгоритм векторизации, основанный на лексических подстановках:

1. Используются прямая и обратная языковая модель. Прямая модель обучалась предсказывать следующее слово на основе левого контекста, обратная – на основе правого. Так, например, для текста «из деревянного лука выстрелил человек» с целевым словом 'лука' прямая модель будет получать на вход «Из деревянного», а обратная «выстрелил человек». Каждая из моделей выдает вероятностное распределение слов, которые могут быть на месте целевого слова. По этим вероятностям выбираются топ k слов для левого и правого контекста отдельно.

2. Затем для каждого набора из k слов l раз сэмплируетя случайное слово, тем самым получается набор из $2l$ слов.

3. Шаг 2 повторяется s раз, для полученных наборов применяется $tfidf$

4. Применяется агломеративная кластеризация для получившихся векторов. Для каждого примера выбирается тот кластер, в котором оказалось наибольшее число представителей.

Такую модель можно улучшить, добавив динамические симметричные шаблоны (T и _) для прямой модели, и (_ и T) для обратной, где на месте 'T' стоит целевое слово, а на месте '_' – то что будет предсказывать модель. Так для прошлого примера входами будут «Из деревянного лука и», «и лука выстрелил человек». Таким образом модель будет видеть целевое слово и будет создавать более подходящие подстановки.

3.3 Трансформеры

Отдельное внимание стоит уделить архитектуре Трансформера[8]. Она является одной из ключевых достижений в области глубокого обучения и нейронных сетей, особенно в области обработки естественного языка.

Основной идеей трансформера является замена классических сверточных и рекуррентных слоев нейронной сети на механизм внимания. Он используется для обработки последовательности входных данных и получения контекстуальных представлений каждого элемента в последовательности.
Для решения задачи извлечения смысла слова с помощью архитектуры трансформера можно воспользоваться идеей подхода через Word2Vec, что было сделано в статье [Slapoguzov A. et al][9]

1. Каждый пример подается на вход RuBERT[10], на выходе получаются эмбеддинги токенов.
2. Удаляются эмбеддинги, относящиеся к знакам пунктуации, предлогам.
3. Конкатенируется среднее эмбеддингов последних 4х слоев RuBERT
4. Применяется алгоритм сжатия размерности данных PCA[11].
5. Применяется алгоритм кластеризации affinity propagation.

4 Описание датасета

Для работы был выбран датасет bts-rnc[1], состоящий из примеров употребления многозначных слов в русском языке. Значения слов определялись с помощью смыслового инвенторя Большого толкового словаря[12], а контексты брались из Национального корпуса русского языка[13]. Датасет разбит на две части: обучающая часть, которую можно использовать для подбора гиперпараметров модели, и тестовая часть, на которой можно оценить ее качество и сравнить с другими решениями.

Таблица 1: Датасет bts-rnc-train

context id	word	gold sense	positions	context
1	балка	1	90-94	... потолочную балку ...
2	балка	1	69-74	... балка обрушилась. ...
...	...	...	...	...
83	балка	2	82-87	... степной балке. ...
...	...	...	...	...

Каждая строка датасета содержит информацию о целевом слове, а именно само целевое слово, его контекст, позиции начала и конца для удобства поиска, и золатая метка (смысл многозначного слова) (см. Таблица 1).

5 Реализация модели

Для своего решение была использована модель XLM-R[14] - трансформер, предобученный на задаче маскированого языкового моделирования более чем для 100 языков. Такой выбор модели обосновывается тем, что эта модель хорошо проявляла себя на различных задачах компьютерной лингвистики, а также объем данных на русском языке, на котором эта модель предобучалась, занимает второе место.

В отличие от ранее описанной работы с RuBERT[10], для векторизации будут использоваться выходы не всего примера, а только целевого слова. Поскольку модель состоит из слоев внимания, векторное представление будет содержать информацию о контексте слова, что и требуется для этой задачи. Однако сразу получить вектор целевого слова не получится. XLM-R работает с токенами, которых в слове может быть несколько.

5.1 Выбор метода работы с токенами

Первая подзадача – подбор метода объединения векторов подслов целевого слова. Простым методом объединения подслов является взятие среднего от их эмбеддингов:

$$subword\ pooler_1(e_1, e_2..e_n) = \frac{e_1+e_2+..+e_n}{n}$$

Однако, такой метод оказался не самым удачным. Впоследствии была рассмотрена токенизация XLM-R и замечен интересный факт: количество символов в первом подслове в среднем больше, чем в остальных подсловах. Например, для предложения «Не упади в пропасть.» XLM-R разбивает слово 'пропасть' на два подслова. Первое подслово состоит из 'пропаст', а второе только из 'ь'. После этого появилась идея использовать не среднее эмбеддингов всех подслов, а взвешенную среднее с дополнительным гиперпараметром w, отвечающий за вес первого подслова:

$$subword\ pooler_2(e_1, e_2..e_n) = \frac{w*e_1+e_2+..+e_n}{w+n-1}, w \in [1, +\infty)$$

При $w = 1$ мы получаем первоначальное среднее всех эмбеддингов, а когда устремляем w к бесконечности, получаем эмбеддинг первого подслова.

При правильном параметре w такой метод работает гораздо лучше предыдущего.
Затем был придуман метод, показывающий наилучший результат:

$$subword\ pooler_3(e_1, e_2..e_n) = e_1 + k * \frac{e_1+e_2+..+e_n}{n}, 0 < k < 1$$

Здесь мы считаем взвешенную сумму первого эмбеддинга и среднего всех эмбеддингов. Гиперпараметр k принимает значения меньше единицы, поскольку мы хотим сделать эмбеддинг первого слова важнее среднего эмбеддингов.

5.2 Объединение выходов слоев модели

Вторым шагом является определение слоев, с которых будут браться выходы XLM-R, а также способ агрегации эмбеддингов. Брать выходы только последнего слоя может привести к плохим результатам, поскольку в процессе предобучения модель настраивалась точно предсказывать пропущенное слово, а значит слова с разными грамматическими формами могут иметь сильно различающиеся эмбеддинги в не зависимости от того, одинаковый ли у них смысл или нет.

Поэтому было решено агрегировать выходы с послетних l слоев, где l - дополнительный гиперпараметр. В качестве агрегации эмбеддингов были выбраны взятие среднего, конкатенация и поэлементный максимум. Далее удалось добиться улучшения за счет отказа от агрегирования. Вместо этого берутся выходы лишь одного слоя с номером l с конца.

5.3 Кластеризация

Последним этапом является кластеризация векторных представлений примеров. В качестве кластеризации была выбрана агломеративная кластеризация, как одна из наиболее удачных, и показавшая хорошие резудьтаты в существующих решениях. Принцип этого алгоритма кластеризации следующий: изначально каждому примеру выделяется свой отдельный кластер. Затем на каждой итерации алгоритма происходит слияние двух кластеров в один по некоторому критерию. Алгоритм прекращает свою работу при достижении

заранее заданного числа кластеров.

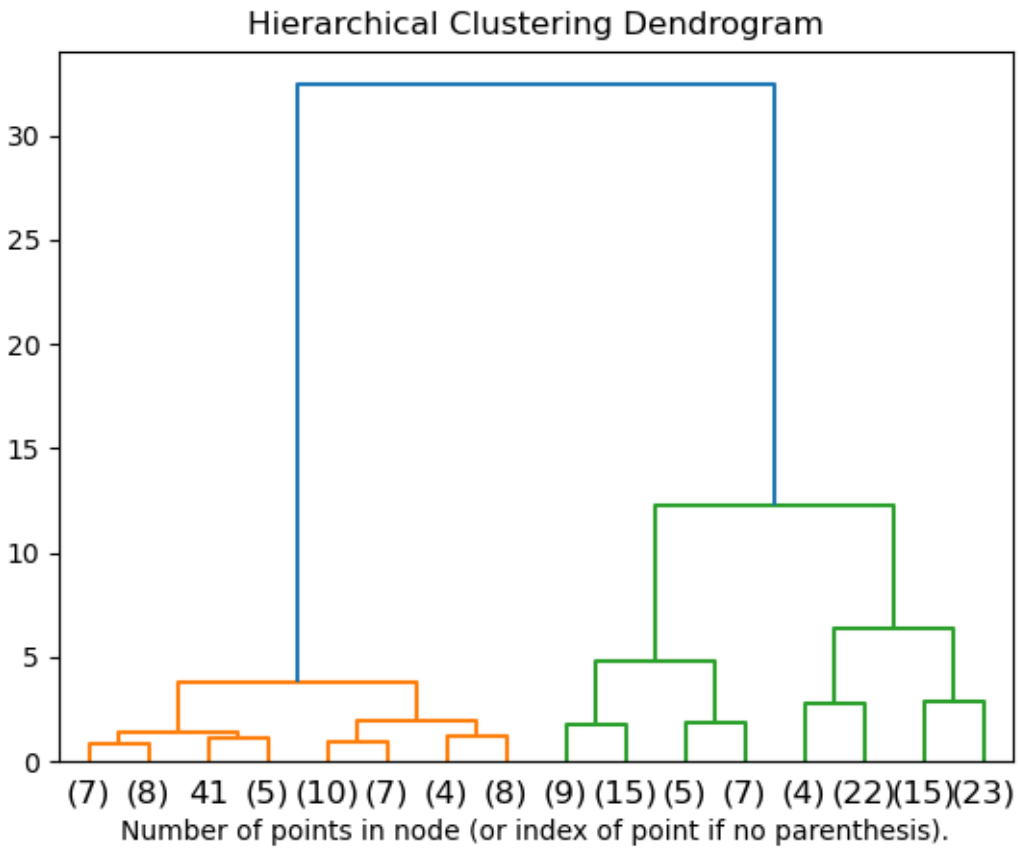

Рис. 1: Пример работы агломеративной кластеризации.

У данной кластеризации есть 3 гиперпараметра, которые необходимо подобрать:

- Число кластеров. Существует 2 популярные стратегии выбора числа кластеров: фиксированное количество кластеров для всех слов или подбор числа по некоторым внутренним метрикам кластеризации. Было принято решение взять число кластеров фиксированным и равным 3, как среднее количество смыслов слов в обучающей выборке.

- Критерий связывания. У агломеративной кластеризации всего 4 критерия связывания: критерии минимизирующие минимальное / среднее / максимальное расстояние между парами объектов двух различных кластеров, и критерий минимизирующий дисперсию объединяющихся кластеров. Экспериментальным путем получилось, что последний критерий справляется со своей задачей лучше всего.

- Метрика. Были перепробованы евклидова метрика, манхэттенское расстояние, косинусная близость. В конце выбор пал на евклидову метрику, поскольку только с ней возможно использовать критерий, минимизирующий дисперсию.

6 Результаты тестирования

После тщательного подбора гиперпараметров модель была запущена на тестовой выборке. В качестве метрики качества модели используется adjusted rand index (ARI)[15]:

$$RI = \frac{TP+TN}{TP+TN+FP+FN},\ ARI = \frac{RI-\mathbb{E}(RI)}{1-\mathbb{E}(RI)}$$

где TP - число пар с одинаковым смыслом и попавшие в один кластер, TN - число пар с разными смыслами и попавшие в разные кластеры, FP - число пар с разными смыслами,но попавшие в один кластер, FN - число пар с одним смыслом, но попавшие в разные кластеры.

Такая метрика хороша тем, что при фиксированом числе кластеров при случайной кластеризации матожидание ARI равно 0:

$$\mathbb{E}(ARI) = \mathbb{E}(\frac{RI-\mathbb{E}(RI)}{1-\mathbb{E}(RI)}) = \frac{\mathbb{E}(RI)-\mathbb{E}(RI)}{1-\mathbb{E}(RI)} = 0$$

В качестве результатов для сравнения были взяты три наилучшие модели, использующих Word2Vec[1], модель на лексических подстановках[16] и RuBERT[10].

Таблица 2: Результаты тестирования

Модель	train ARI	test-public ARI	test-private ARI
Word2Vec (top 1)		0.35	0.33
Word2Vec (top 2)		0.28	0.28
Word2Vec (top 3)		0.24	0.24
Lexical Substitutions		0.50	0.45
RuBERT		0.21	0.21
XLM-R	0.34	0.24	0.25

По результатам (см. Таблица 2), модель обошла RuBERT. Однако XLM-R получилась ни чуть не лучше Word2Vec, и сильно проигрывает лексическим подстановкам. Также заметно большое падение ARI при переходе от обучающей выборки к тестовой. Это означает, что модель плохо работает с новыми словами.

7 Заключение

Результатом данной работы является следующее:

1. Рассмотрены различные способы получения векторных представлений примеров употребления многозначных слов.

2. Построена модель для задачи извлечения смысла слова и подобраны оптимальные гиперпараметры.

3. Проведено сравнение модели с существующими решениями.

8 Список Литературы

[1] RUSSE'2018: A Shared Task on Word Sense Induction for the Russian Language [Электронный ресурс]
url - https://www.dialog-21.ru/media/4539/panchenkoaplusetal.pdf
(дата обращения 10.05.2023).

[2] Efficient estimation of word representations in vector space [Электронный ресурс] url - https://arxiv.org/abs/1301.3781 (дата обращения 10.05.2023).

[3] tfidf [Электронный ресурс] url - https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
(дата обращения 10.05.2023).

[4] chisq [Электронный ресурс] url - https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chisquare.html
(дата обращения 10.05.2023).

[5] Агломеративная кластеризация [Электронный ресурс]
url - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html (дата обращения 10.05.2023).

[6] Affinity propagation [Электронный ресурс]
url - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html (дата обращения 10.05.2023).

[7] Asaf Amrami, Yoav Goldberg *Word Sense Induction with Neural biLM and Symmetric Patterns* //In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium - 2018. pages 4860–4867

[8] Attention is all you need [Электронный ресурс]
url - https://arxiv.org/abs/1706.03762 (дата обращения 10.05.2023).

[9] Slapoguzov A *et al. Word Sense Induction for Russian Texts Using BERT* //Proceedings of the 28th FRUCT conference - 2021. pages 621-627.

[10] RuBERT [электронный ресурс]
url - https://huggingface.co/DeepPavlov/rubert-base-cased
(дата обращения 10.05.2023).

[11] PCA [Электронный ресурс]
url - https://scikit-learn.org/stable/modules/generated/
sklearn.decomposition.PCA.html (дата обращения 10.05.2023).

[12] Большой Толковый Словарь Русского Языка / Гл. Ред. Кузнецов С.А. СПб.: Норинт, 2000. - 1536 с.

[13] Национальный корпус русского языка [Электронный ресурс]
url - https://ruscorpora.ru (дата обращения 10.05.2023).

[14] Unsupervised Cross-lingual Representation Learning at Scale [Электронный ресурс] url - https://arxiv.org/abs/1911.02116
(дата обращения 10.05.2023).

[15] Hubert L., Arabie P. *Comparing Partitions* // Journal of Classification, Vol. 2, № 1 - 1985. pages 193–218

[16] Arefyev N. *et al. Combining Neural Language Models for WordSense Induction* // International Conference on Analysis of Images, Social Networks and Texts AIST 2019: Analysis of Images, Social Networks and Texts, pages 105-121

9 Приложение А. Подбор гиперпараметров.

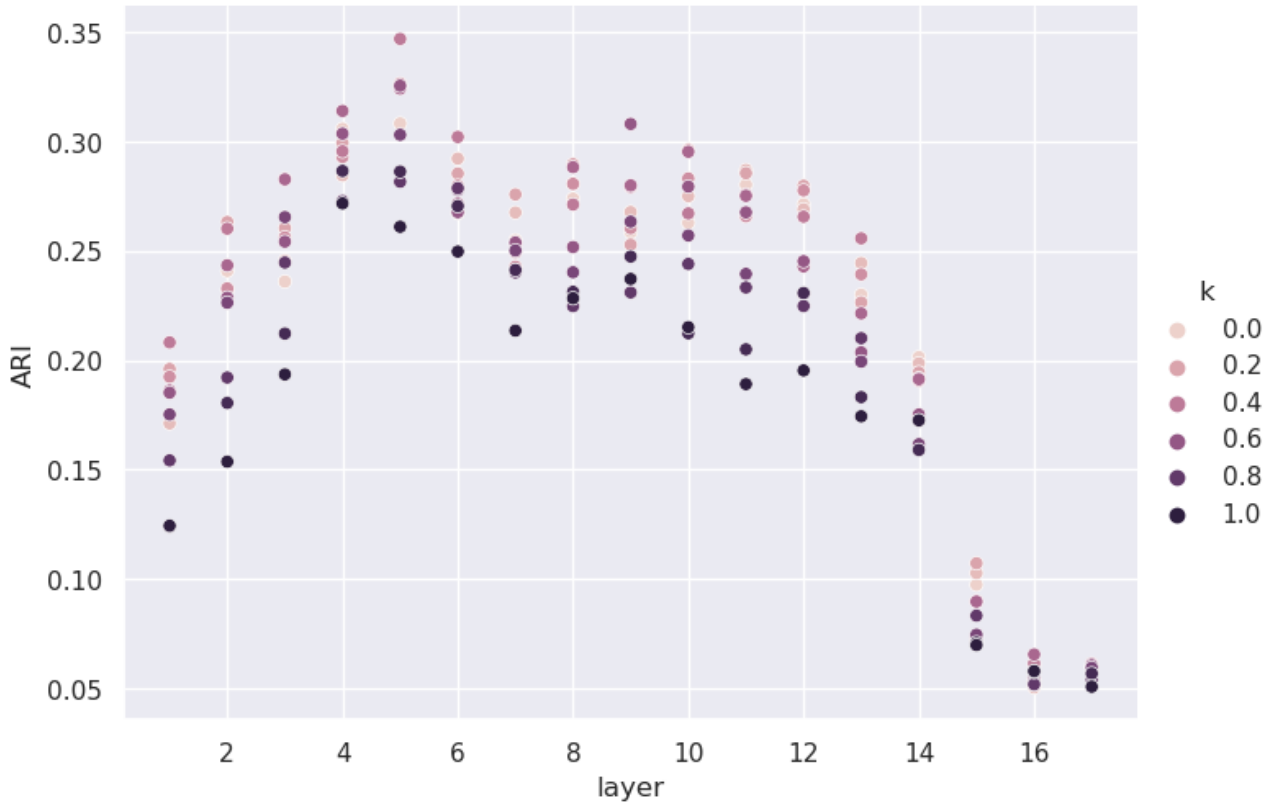

Рис. 2: Финальный перебор гиперпараметров. Используется *subword* $pooler_3$, выходы берутся только со слоя с номером *layer* с конца.