

Санкт-Петербургский государственный университет
Факультет Прикладной математики - Процессов управления

Кафедра Технологии программирования

Коробков Никита Александрович
Выпускная квалификационная работа бакалавра

Метод межъязыковой адаптации диалоговых систем

Направление 01.03.02
Прикладная математика и информатика

Научный руководитель:
старший преподаватель Мишенин А.Н.

Санкт-Петербург
2019

Оглавление

Введение	2
Постановка задачи	5
Обзор литературы	7
1. Обработка данных	10
2. Алгоритм	12
2.1. Подбор Модели	12
3. Результаты	13
Выводы	14
Заключение	15
Список литературы	16

Введение

В последние годы все больше внимания уделяется решению различных задач с применением машинного обучения и нейронных сетей. Одним из наиболее интересных направлений в области применения современных нейросетевых методов являются диалоговые системы.

Диалоговая система это набор программ и алгоритмов позволяющих человеку вести диалог с программой в манере свойственной человеческой. Иногда диалоговые системы называют разговорным искусственным интеллектом или "чат-ботом".

Задача построения подобных программ является актуальной для промышленной области потому, что решение задач пользователей путем диалога с агентом поддержки всегда было и будет оставаться наиболее простым и эффективным методом. Так как вербальное общение наиболее естественный для человека способ коммуникации. Построение хорошей и надежной диалоговой системы заменяющей работников колл центра позволило бы существенно сократить затраты ресурсов.

Диалоговые системы можно разделить на системы общего назначения и задачеориентированные. Чем более узкую задачу призвана решать система, тем проще она может быть устроена. В самом тривиальном случае система может просто возвращать заранее известный ответ. Например текущее время.

Мы хотели бы построить систему решающую более общую задачу. Для этого в нужно сначала выяснить детально чего именно хочет пользователь (задать вопрос и получить ответ). В самой тривиальной реализации этот сценарий может выглядеть как простой выбор задачи из списка. С применением такого подхода в промышленности можно столкнуться уже сегодня. Позвонив в банк и авиакомпанию можно

часто услышать робота, который предложит нажать разные кнопки в зависимости от цели звонка ("Для проверки баланса нажмите 1, для уточнения статуса заявки нажмите 2" и т.п.). Этот подход хорош тем, что не требует от системы никакой интеллектуальности и работает очень надежно. Однако занимает много пользовательского времени и зачастую нервирует. В идеальном случае мы хотели бы получив запрос в виде предложения на натуральном языке, например "Какая погода сейчас на улице?" сразу распознать намерение пользователя.

Это одна из подзадач в построении диалоговых систем, на которой хотелось бы сконцентрировать наше внимание в этой работе. Данная задача является довольно актуальной и стоит уже давно. Так что для ее решения было предложено множество методик. Подобнее они будут рассмотрены в разделе "Обзор литературы". Большинство методов ориентированно на работу с английским языком. В основном потому, что для английского собранно наиболее большое количество данных. Кроме того английский просто считается языком по умолчанию в научной среде.

В то время как для английского языка достигнуты внушительные результаты, ситуация с другими языками обстоит несколько иначе. Представленные модели в большинстве своем обучены на колоссальном объеме размеченных данных. Таких данных не существует для более редких языков. По этому простое перенесение достигнутых результатов путем обучения идентичной модели на другом языке не представляется возможным. Тем не менее, почти для всех мировых языков существуют полные словари. Пользуясь общими знаниями о связи двух языков (словари, параллельные тексты) можно обобщить знания одной модели на другой язык не используя размеченных данных для второго языка совсем (либо используя совсем немного). Данный подход в литературе носит название Transfer Learning. Целью данной работы будет построение модели для извлечения намерения

из предложения на **шведском** языке при помощи переноса знаний накопленных обученной на английском языке модели. Выработанную методику можно будет использовать для построения моделей приблизительно такой же точности для любого другого языка для которого существует словарь перевода слов на английский.

Постановка задачи

Пусть существует множество команд на английском языке E и конечное множество намерений K

$$K = \{k_1, k_2, \dots, k_n\}$$

Каждой команде из E однозначно соответствует элемент множества K . Соответствие обозначим I_e

$$I_e : E \rightarrow K$$

Тренировочные данные состоят из множеств E, K и соответствия I_e . При этом существует так же функции-переводчик T , которая каждой команде на английском языке ставит в соответствие команду на шведском языке. Множество команд на шведском языке обозначим S

$$T : E \rightarrow S$$

Целью данной работы будет получение функции $I_s : S \rightarrow K$ сопоставляющей любой команде на шведском языке намерение. При этом должны выполняться два условия.

1. Намерение должно совпадать с намерением перевода шведской команды на английский язык.

$$\forall s \in S \quad I_s(s) = I_e(T^{-1}(s)) \quad (1)$$

$$\forall e \in E \quad I_s(T(e)) = I_e(e) \quad (2)$$

2. Функция I_s не должна зависеть от функции T

Второе условие является определяющим для данной задачи. Если бы мы могли использовать функцию T в I_s то можно было бы просто определить I_s как в (1) и остановиться на этом. Но это невозможно,

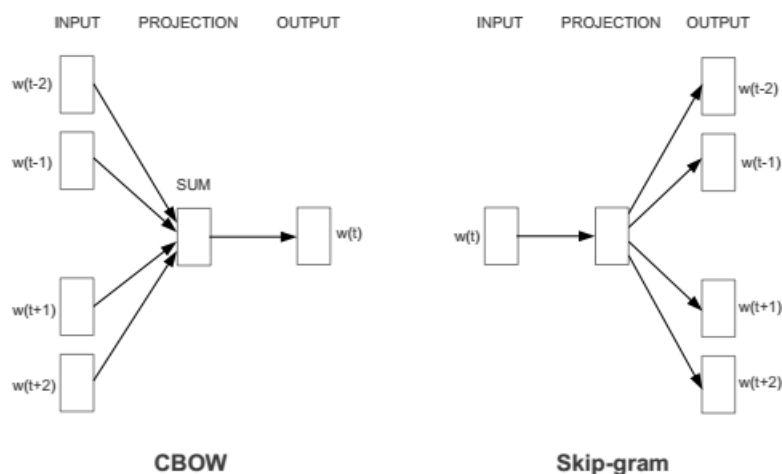
так как перевод (вычисление T^{-1}) это слишком дорогостоящая операция. Мы бы хотели получить функцию которая была бы достаточно легковычисляемой для использования в мобильных приложениях. Поэтому вместо прямого перевода команды со шведского языка на английский и последующего применения существующих алгоритмов мы постараемся выделить какие-то ключевые атрибуты команды на шведском языке и использовать их для распознавания намерения.

В силу требования 2 мы едва ли сможем удовлетворить требование 1 полностью. Вместо этого попытаемся построить систему, которая на тренировочных данных сможет максимально часто предсказывать намерения пользователя правильно. В качестве метрики качества предсказаний будем использовать долю команд в тестовой выборке, по которой система приняла правильное решение.

Обзор литературы

Задача построения диалоговых систем лежит в области автоматического анализа текстовых данных. Одним из главных вопросов применения нейронных сетей в данной области является эффективное представление слов в памяти компьютера. При анализе текстов мы бы хотели заменять слова на вектора как-то отражающие семантический смысл слова. Наиболее распространенный метод получения векторов слов описан в работах [9]. В данном подходе обучающий текстовый корпус просматривается окном ширины $2h + 1$ слов, и для каждого окна однослойная нейронная сеть предсказывает центральное слово окна $w(t)$ по окружающим $w(t + i)$, $i \in [-h, h]$ или наоборот. Эти архитектуры называются Continuous Bag-of-words и Skipgram соответственно. Минимизируя ошибку предсказания, нейронная сеть строит проекцию слов в векторное пространство заранее определенной размерности. При достижении заданной точности предсказания или определенного числа эпох, алгоритм генерирует словарь с векторными представлениями для слов из обучающего корпуса.

Рис. 1. Архитектуры нейронных сетей, представленные в citew2v для окна ширины $h = 5$



Данный подход позволяет получить вектора обладающие свойством семантической близости. Мы можем надеяться что слова обладающие схожим смыслом будут находится рядом в построенном векторном пространстве. После того как методы описанные в [9] показали свою эффективность в ряде задач обработки текстов [10]. Было разработано и предложено несколько похожих методов построения векторов слов [12], [11]

В данный момент большинство методик обработки естественного языка так или иначе использует вектора слов.

В нашей работе мы использовали дополненную реализацию оригинального word to vec "fasttext" [1]. В отличие от оригинальной архитектуры этот подход помимо слов контекста использует части слов для обучения, что позволяет предсказывать вектора слов для слов отсутствующих в тренировочной коллекции.

Для работы с последовательностями слов (предложениями) часто применяются рекуррентные нейронные сети описанные в статьях [13] [7]. Воизбежание проблемы затухающих градиентов при обучении, используют LSTM архитектуру [6].

В статье Attention is all you need [14] группа исследователей из Google описывает принципиально новый подход к обработке последовательной текстовой информации и в частности к переводам. Вместо классической архитектуры рекуррентных нейронных сетей с использованием LSTM или GRU модулей автор использует так называемый механизм внимания, Который позволяет более качественно представлять информацию содержащуюся в предложениях на этапе кодирования. Такую сеть так же называют "Трансформер" из за гибкости внутренней структуры, позволяющей получать разную информацию о кодируемом предложении в зависимости от запроса. Многие современные автоматические переводчики пользуются этой технологией. В нашей работе мы пользовались готовой системой автоматическо-

го перевода от Яндекс. Примерное описание механизмов работы их переводчика доступно в статье [15]

Идея трансформер сетей развивается в статье [4]. Авторы предлагают тренировать многослойную модель из трансформер модулей на задаче определения связности предложений и предсказания пропущенного слова. Полученная модель показывает исключительные результаты после дообучения на ряде конкретных задач.

Задача предсказания намерения из фиксированного множества может быть сформулированная как задача классификации предложений. Интересный подход к этой задаче с использованием сверточных сетей предложен в статье [8].

1. Обработка данных

В нашей работе в качестве тренировочных и тестовых данных мы использовали датасет собранный компанией SNIPS [3]. Данные включают в себя более 13000 запросов пользователей на английском языке. Каждый запрос относится к одной из семи категорий по намерению пользователя.

Представленные намерения:

- Get weather (Посмотреть погоду - 2000 записей)
- Play music (Включить музыку - 2000 записей)
- Book restaurant (Забронировать ресторан - 1973 записей)
- Search creative work (Посик произведений - 1954 записей)
- Add to playlist (Добавить в плейлист - 1942 записей)
- Rate book (Поставить оценку книге - 1956 записей)
- Search movie schedule (Расписание сеансов кино - 1959 записей)

Данные предоставлены компанией SNIPS по лицензии Creative Commons Zero v1.0 Universal и доступны для скачивания по ссылке <https://github.com/snipsco/nlu-benchmark>

Таблица 1. Примеры запросов из датасета SNIPS

Категория	Пример
Get weather	What is the forecast at 12 am in Sudan.
Play music	Play some 1954 songs on my Itunes.
Book restaurant	What is the forecast at 12 am in Sudan.
Search creative work	Find me the Lace and Whiskey soundtrack.
Add to playlist	Add this artist to spring music.
Rate book	give this textbook a 5 out of 6 rating.
Search movie schedule	Where is Road to the Stage playing.

Для подготовки тренировочных и тестовых данных на шведском языке мы использовали API сервиса Яндекс Переводчик ¹.

¹<https://translate.yandex.ru/developers>

2. Алгоритм

2.1. Подбор Модели

Для того чтобы изучать перенос знаний модели для английского языка на другой, нужно сначала подобрать хорошую архитектуру, работающую в одном домене.

Такая архитектура с одной стороны должна быть достаточно простой, для того чтобы ее можно было разбить на части и транслировать на другой язык, а с другой показывать хорошие результаты.

В нашей работе мы рассматривали модели основанные на векторах слов или векторах предложений.

- **FastText - average** – для каждого слова из предложения запроса вычислялся его вектор с использованием технологии FastText [5]. Полученные вектора усреднялись и средний вектор использовался для обучения линейной регрессии. Размерность вектора запроса - 300.
- **ELMo - average** – для каждого слова из предложения запроса вычислялся его вектор с использованием технологии ELMo [12]. Полученные вектора усреднялись и средний вектор использовался для обучения линейной регрессии. Размерность вектора запроса - 1024.
- **Universal Sentence Encoder** – Каждый запрос кодировался в вектор длины 512 при помощи технологии USE [2]. Полученные вектора использовались для обучения линейной регрессии.

3. Результаты

Результаты

Выводы

Выводы

Заклучение

Заклучение

Список литературы

- [1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.
- [2] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Stroe, and Ray Kurzweil. Universal sentence encoder. *CoRR*, abs/1803.11175, 2018.
- [3] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190, 2018.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.
- [5] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- [6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
- [7] Andrej Karpathy, Justin Johnson, and Fei-Fei Li. Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078, 2015.

- [8] Yoon Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.
- [9] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *ArXiv e-prints*, January 2013.
- [10] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting Similarities among Languages for Machine Translation. *ArXiv e-prints*, September 2013.
- [11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.
- [12] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.
- [13] Alex Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314, 2018.
- [14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.
- [15] Yandex. Технологии - Машинный перевод, 2019. Дата обращения - 4 апреля 2019.