Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математике и кибернетики
Кафедра математических методов прогнозирования

Построение обобщенного графа диалога

КУРСОВАЯ РАБОТА

Выполнил:
студент 3 курса 317 группы
Штыков Павел Дмитриевич

Научный руководитель:
д.ф.-м.н., профессор
А. Г. Дьяконов

Москва, 2022

Содержание

1 Введение

Обработка естественного языка (NLP) является ключевой задачей в машинном обучении, а обработка диалогов является важной ее частью. Одной из слабо изученных областей в обработке диалогов является проблема построения и представления общей структуры диалога.

Естественно предположить, что у диалогов из одной области может быть некоторая общая структура. Так же естественно представлять эту структуру в виде графа. Такой граф позволяет представить информацию о корпусе однородных диалогов в сжатой форме, подходящей как для визуализации, так и для встраивания в более сложные диалоговые системы. В данной работе мы предложим формализацию понятия обобщенного графа диалога и базовые способы его построения и визуализации.

2 Существующие подходы

К сожалению, нам не удалось найти большое число работ связанных с данной темой. Существует серия статей [14], [10], в которой предлагается два способа построения графа диалога. Первый основан на использовании вариационных автокодировщиков в паре с рекуррентной нейронной сетью [4]. Во второй статье авторы добавили в свою архитектуру механизм внимания [17], что позволило им улучшить качество. Результат их работы можно увидеть на Рис. 1.

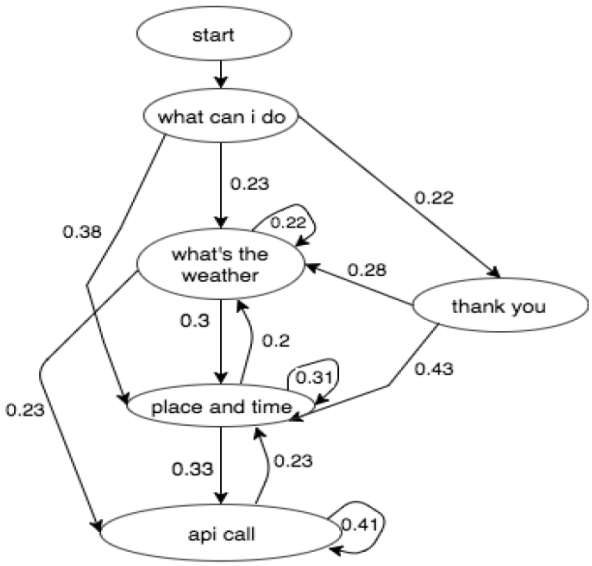
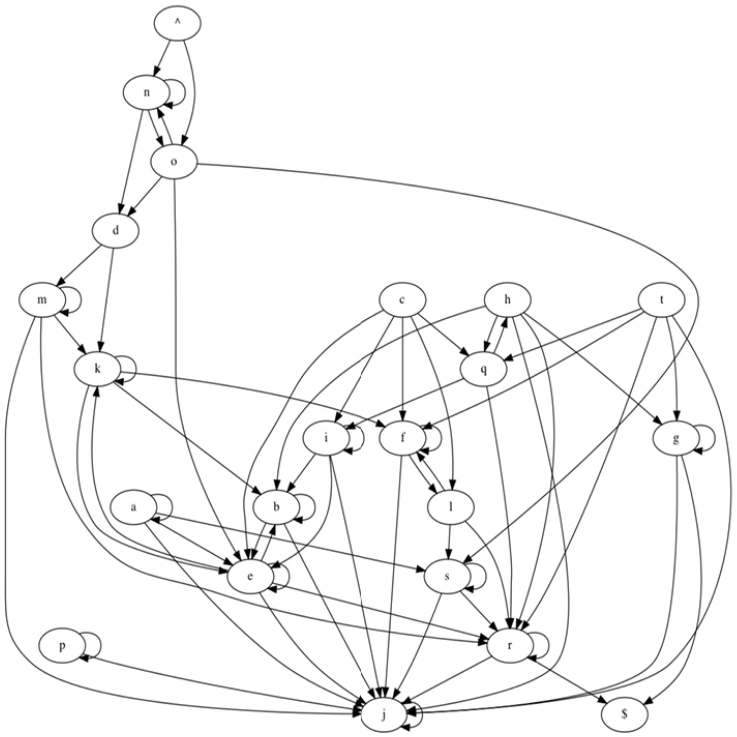


Рис. 1: Пример обобщенного графа диалога взятого из статьи [14]

В своей же работе мы будет ориентироваться на следующую более работу [9]. В ней авторы для построения графа применяют алгоритм классификации SCAN [16], работающий без учителя. Данный алгоритм может кластеризировать тексты и дополнительно разметить их. Для эмбеддинга текстов авторы использовали стандартный BERT [6]. В нашей работе мы исследуем применимость эмбеддинга, более подходящего для семантической кластеризации — SBERT [11]. Граф полученный авторами представлен на рис. 2.

Во всех упомянутых выше статьях авторы сравнивают свои подходы с методом k-средних, однако не приводят самого алгоритма построения графа диалога с помощью простой кластеризации. В данной работе мы предложим пример такого алгоритма. Также мы приведем уточнения определения обобщенного графа диалога. На наш взгляд, граф, соответствующий такому определению, проще в дальнейшем анализировать и визуализировать.



cluster-name	manual-label
^	start-node
n	self-introduction
o	rpc-inquiry
d	brand-intro
m	product-info
k	payment-inquiry
f	amount-info
l	date-reminder
r	number-intimation
g	payment-date-inquiry
$	end-node

Рис. 2: Пример обобщенного графа диалога взятого из статьи [9]

3 Постановка задачи

Введем определение обобщенного графа диалога.

Определение 1. *Назовем обобщенным графом диалога пару* $T = (G, p(u|v))$*, где:*

- $G = (V, E)$ *— направленный взвешенный граф. С каждым ребром в графе* G *ассоциирована вероятность перехода по этому ребру:* $e_{i,j} \sim p(v_i|v_j)$*. При этом сумма вероятностей ребер выходящих из каждой вершины равна* 1*:* $\sum_j p(v_i|v_j) = 1$

- $u \in U$ *— единичное высказывание, а* U *— пространство всех высказываний во всех диалогах;*

- $p(u|v)$ *— плотность вероятности (либо функция вероятности в случае дискретного пространства* U*) отнесения высказывания* u *к текущей вершине* v*.*

Такое определение не ограничивает нас в выборе модели для его построения. Дополнительное требование наличия функции $p(u|v)$ позволит нам вычислять статистики полезные для визуализации и дальнейшего использования графа (например самое вероятное предложение или самые частотные слова среди предложений ассоциированных с текущей вершиной).

Также такой граф достаточно просто обобщается на случай персонализированных диалогов (например диалога *«пользователь»-«система»*) — введением раскраски вершин, т.е. дополнительной функции $\phi(v)$, ставящей в соответствие каждой вершине некоторый персональный идентификатор пользователя (ID). Однако необходимо ввести дополнительные ограничения. Так как высказывания пользователей чередуются, то логично потребовать, чтобы вершины разных цветов не были инцидентны. Также основное определение не запрещает петли. В персонализированном графе их стоит запретить. В данной работе мы будем строить простой граф диалога, без раскраски.

Дополнительно, пусть $D = \{d_1, d_2, \ldots, d_{|D|}\}$ — выборка диалогов, где d_i — один диалог. Каждый диалог является набором из нескольких высказываний: $d_i = \{d_i^1, d_i^2, \ldots, d_i^n\}$, $d_i^j \in U$. В данной работе мы будем работать с неразмеченными диалогами. Однако, в общем случае нет ограничения на использование разметки.

Добавим к каждому диалогу x_i технические высказывания *начала* и *конца* диалога ($BEGIN$ и END). Аналогичные вершины добавим и в граф. Вероятность $p(u|v)$ для этих вершин будет вырождена в соответствующих точках в пространстве высказываний U. Это необходимо для более ясной конструкции графа и соблюдения ограничения на сумму вероятностей ребер, исходящих из вершины: $\sum_j p(v_i|v_j) = 1$.

4 Предложенный метод

Для более простой и понятной работы с высказывания перейдем в промежуточное пространство — воспользуемся эмбеддингом (*embedding*):

$$Embedding : U \to M$$

В качестве эмбеддинга мы использовали предобученную сиамскую нейронную сеть [11] с разными базовыми сетями (подробнее в разделе 5). В дальнейшем, если не оговорено другого, под высказыванием u мы будем подразумевать его эмбеддинг $Embedding(u)$.

Пространство M метрическое (в нашем случае это $\mathbb{R}^{768}$) с косинусной метрикой, отражающую семантическую близость высказываний, следовательно в нем можно воспользоваться метрическими метода кластеризации.

Теперь приведем алгоритм построения обобщенного графа диалога T, для высказываний в пространстве эмбеддингов M.

▶ Пусть существует некоторый алгоритм кластеризации a:

$$a : M \to V$$

В данном случае множество вершин V есть множество кластеров. Соответственно может быть вычислена *дискретная* вероятность принадлежности каждого высказывания к каждой вершине:

$$a(u) = p(v|u)$$

При этом кластеризация может быть как жесткой (например методом k-средних (*k-means*)), так и мягкой (например смесью гауссиан (*GMM*)).

Зная $p(v|u)$, можно вычислить $p(u|v)$, используя теорему Байеса:

$$p(u|v) = \frac{p(v|u)p(u)}{\sum_{i=1}^{|U|} p(v|u_i)p(u_i)},$$

где $p(u)$ частота встречаемости высказывания u во всем корпусе диалогов. Заметим, что вероятность $p(u)$ не одинакова для всех высказываний, так как в корпусе могут встречаться диалоги с одинаковыми высказываниями.

Нам осталось построить в графе ребра и найти вероятности, ассоциированные с ними. Введем в пространстве высказываний U граф $\hat{G}$, подобный графу G, т.е. ориентированный взвешенный граф с вероятностями, ассоциированными с ребрами:

$$\hat{G} = (\hat{V}, \hat{E}), \quad \hat{V} \subset M, \quad \hat{E} \subset \hat{V} \times \hat{V}, \quad \hat{e}_{i,j} \sim p(u_j|u_i).$$

Данный граф строиться напрямую по выборке диалогов, и ребра в нем имеют смысл апостериорной вероятности встретить ответ u_j на высказывание u_i. Соответственно матрица смежности $\hat{A}$ графа $\hat{G}$ определяется как:

$$\hat{A} = (\hat{a}_{ij}), \quad \hat{a}_{ij} = p(u_j|u_i), \quad i, j = \overline{1, |U|}$$

Аналогично матрица смежности вводиться и для основного графа G:

$$A = (a_{ij}), \quad a_{ij} = p(v_j|v_i), \quad i, j = \overline{1, |V|}$$

Так как в нашем случае совместные распределения $p(u|v)$ и $p(v|u)$ дискретны, то они могут быть представлены в виде матриц. Следовательно матрица смежности A получается простым перемножением трех матриц:

$$A = p(u|v) \cdot \hat{A} \cdot p(v|u).$$

Мы закончили построение обобщенного графа диалога T. ■

Заметим, что данный алгоритм построения обобщенного графа применим не только в случае использования кластеризации в пространстве эмбеддингов, но и в случае использования любого другого алгоритма способного оценить апостериорные вероятности $p(v|u)$ (например с помощью латентного размещения Дирихле (*LDA*) или нейронной сети, решающей задачу от начала до конца без промежуточного использования эмбеддингов). Кластеризация была выбрана, как наиболее простой метод.

5 Эксперименты

Нам не известен датасет полностью соответствующий поставленной задаче, т. е. датасет, состоящий из диалогов, имеющих некоторую общую *известную* структуру. Поэтому для проведения экспериментов нами было выбрано два стандартных датасета, для которых, на наш взгляд, можно предположить наличие общей структуры. Первый датасет — *Customer Support on Twitter* [1], в котором собраны ответы официальных аккаунтов технической поддержки крупных американских компаний. Мы выбрали из него подмножество сообщений аккаунтов шести разных авиакомпаний, чтобы сделать датасет более однородным. Второй датасет — *DailyDialog* [7], в котором собраны обычные диалоги из повседневной жизни на разные темы. Для экспериментов мы взяли диалоги на тему работы, как наиболее однородные и узкие. Примеры диалогов из обоих датасетов приведены на рис. 3.

Twitter Customer Support

— @AlaskaAir it says you open at 5:15 @317258 where is everyone? hellooooo https://t.co/WePfUANLsZ
— @429415 @317258 Ticket counter opens at 615 is what I see on our website.
— @AlaskaAir @317258 all good! They just showed up thanks Andre
— @429415 That is good news

DailyDialog

— Everything's gone wrong.
— I know, it's not as I had planned.
— What are we going to do now?
— I'll speak to Bob, he'll be able to help us

Рис. 3: Примеры диалогов из датасетов Twitter Customer Support и DailyDialog

Для начала рассмотрим визуализацию пространства эмбеддингов (Рис. 5 и 4). Мы понизили размерность с помощью t-SNE с перплексией равной 50. Для диалогов из обоих да-

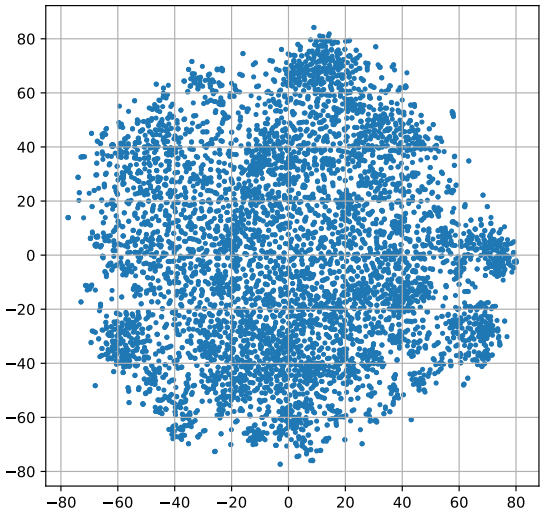


Рис. 4: Пространство эмбеддингов для датасета DailyDialog

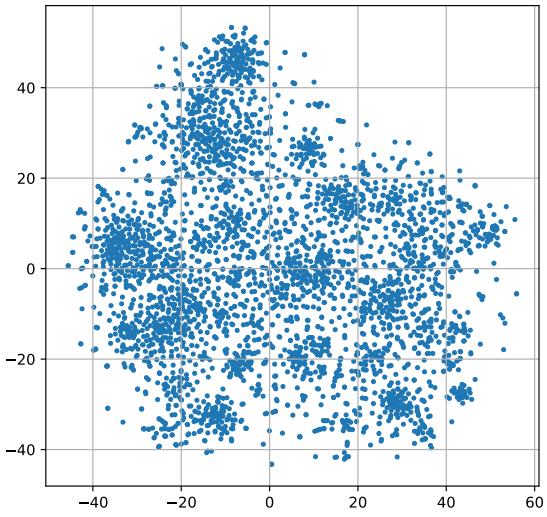


Рис. 5: Пространство эмбеддингов для датасета Twitter Customer Support

тасетов заметна кластерная структура. Это дает нам некоторое подтверждение наличия общей структуры у корпуса диалогов. Однако кластеры небольших размеров и между ними много шума, это может привести к отсутствию четких границ между вершинами графа.

Измерим качество кластеризации. В таблице 5 переставлены результаты работы алгоритма в зависимости от следующих параметров:

- Базовая модель в SBERT: *MPNet* [15], *DistillRoBERTa* [8], [13] и *MiniLM* [18]
- Кластеризатор: k-средних (*KMeans*) и смесь гауссиан (*GMM*)
- Количество кластеров: 5, 10, 15

В качестве метрик использовались следующие базовые метрики качества кластеризации на неразмеченных данных: коэффициент силуэта (*Silh.*) [12], индекс Калински-Харабаса (*C.-H.*) [3] и индекс Дэвиса-Болдина (*D.-B.*) [5].

Введем дополнительную метрику качества для структуры графа. Нам хотелось бы, чтобы граф был более определенным, т.е. было больше ребер с большой вероятностью и меньше ребер с меньшей вероятностью. Для этого будем измерять среднюю нормализованную

энтропию (*Entr.*):

$$H(G) = \frac{1}{|V|}\sum_{i=1}^{|V|}\frac{\sum_{j=1}^{|V|} p(v_j|v_i)}{\log|V|}$$

Соответственно, чем меньше энтропия, тем более определенный граф.

Измерения проводились на предобработанных стандартным образом датасетах (лемматизация, выбрасывание стоп-слов и т.д.).

		Twitter Customer Support				**DailyDialog**			
	Model	Silh.	C.-H.	D.-B.	Entr.	Silh.	C.-H.	D.-B.	Entr.
$\|V\| = 5$	MPNet_KMeans	**0.052**	1043.9	3.912	0.535	0.011	**292.3**	5.166	0.712
	RoBERTa_KMeans	0.043	1003.3	4.43	0.606	**0.024**	281.8	5.176	0.719
	MiniLM_KMeans	0.045	**1054.0**	3.869	0.523	0.023	286.5	5.287	0.724
	MPNet_GMM	0.036	940.4	4.152	0.524	-0.001	253.3	5.517	0.694
	RoBERTa_GMM	0.034	988.7	4.544	0.617	0.022	278.1	5.156	0.71
	MiniLM_GMM	0.044	1046.8	**3.856**	**0.519**	0.01	273.4	**4.74**	**0.663**
$\|V\| = 10$	MPNet_KMeans	0.04	672.1	4.018	0.659	0.016	**210.4**	4.422	0.778
	RoBERTa_KMeans	0.041	634.1	4.147	0.675	**0.021**	195.1	4.394	0.758
	MiniLM_KMeans	**0.054**	**693.4**	3.827	0.668	0.015	198.3	4.482	0.759
	MPNet_GMM	0.037	652.9	**3.669**	**0.597**	-0.002	195.5	4.899	0.776
	RoBERTa_GMM	0.036	617.6	3.992	0.611	0.018	185.1	4.724	0.767
	MiniLM_GMM	0.026	668.6	3.807	0.623	0.009	182.8	**4.373**	**0.73**
$\|V\| = 15$	MPNet_KMeans	0.038	530.2	3.704	0.659	0.018	**167.7**	4.329	0.796
	RoBERTa_KMeans	0.04	490.5	3.936	0.651	**0.023**	155.2	4.27	0.785
	MiniLM_KMeans	**0.049**	**533.7**	3.781	0.668	0.018	160.8	4.234	0.788
	MPNet_GMM	0.015	508.8	3.711	0.626	0.003	158.4	4.404	0.781
	RoBERTa_GMM	0.014	466.8	3.79	**0.623**	0.016	150.7	**4.14**	**0.772**
	MiniLM_GMM	0.03	509.1	**3.663**	0.637	0.014	155.6	4.341	0.775

Таблица 1: Результаты сравнения качества кластеризации для двух датасетов: Twitter Customer Support и DailyDialog

Наконец построим и визуализируем графы. Для всех графов использовалась лучшая модель для данного количества вершин и для данного датасета. В качестве маркировки вершин будем использовать 4-е слова с самым большим значением Tf-Idf. Tf-Idf представления строились для двухсот наиболее вероятных для данной вершины высказываний. Чтобы не засорять рисунок значениями вероятности ребер, мы заменили их разной толщиной ребер: чем толще ребро, тем больше вероятность перехода по нему. Также были убраны ребра с вероятность меньше 0.1: $p(u_j|u_i) < 0.1$. Визуализация графов производилась с помощью пакета GraphViz [2].

На рис. ?? и ?? представлены графы с 5-ю вершинами, составленные по обоим датасетам. Графы для 10-и и 15-и вершин находятся в приложении А.

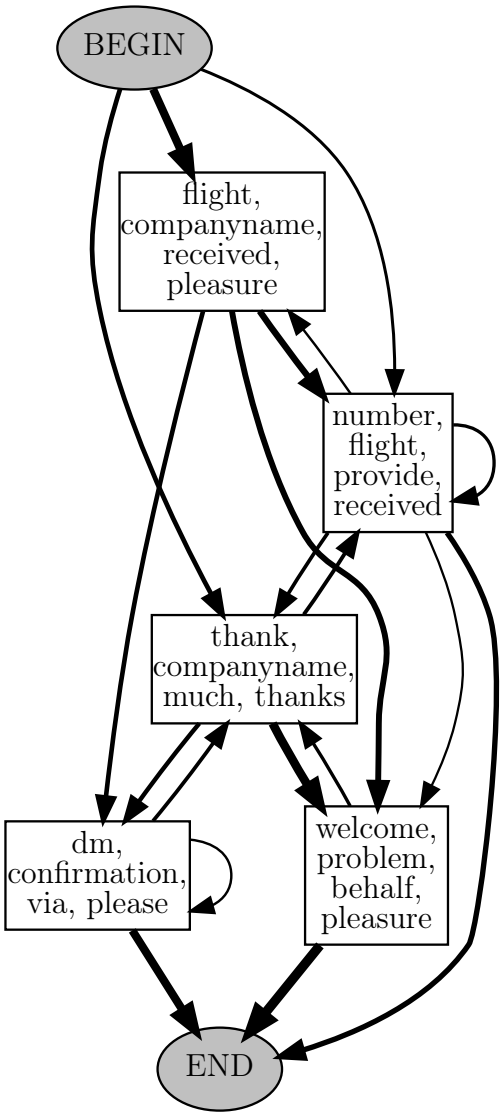


Рис. 6: Граф диалога с 5-ю вершинами для датасета Twitter Customer Support

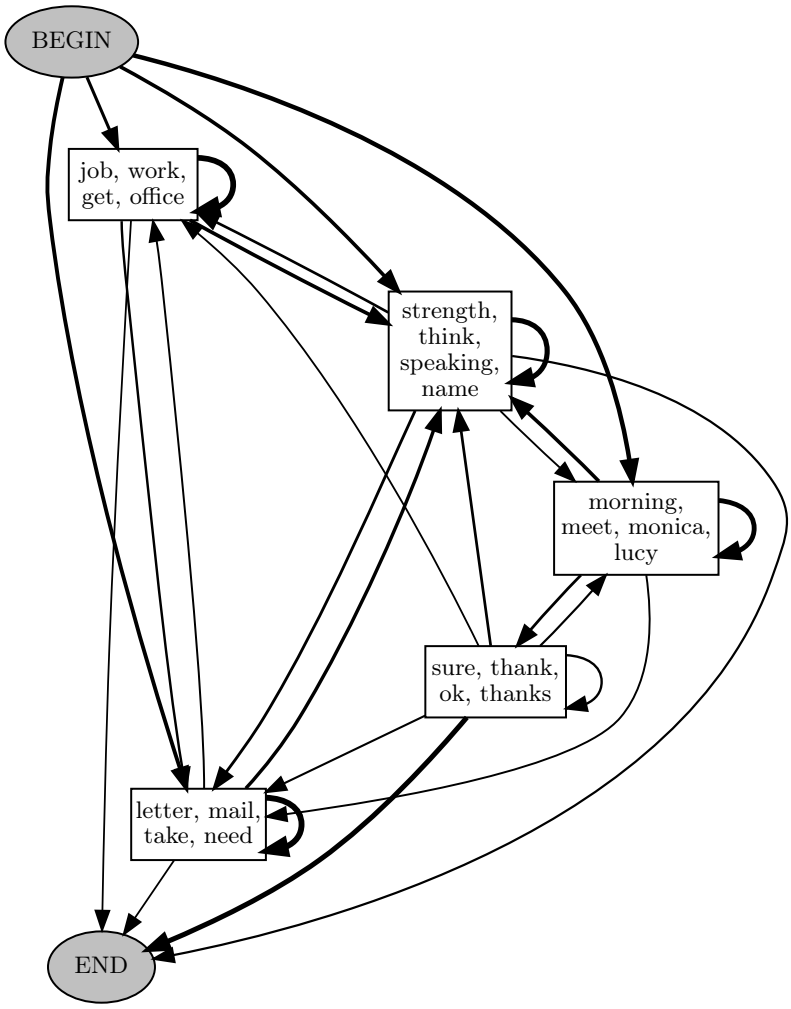


Рис. 7: Граф диалога с 5-ю вершинами для датасета DailyDialog

6 Заключение и будущая работа

В данной работе мы предложили простой алгоритм построения графа диалога с помощью кластеризации в пространстве эмбеддингов SBERT. Также мы уточнили определение самого графа диалога. Однако существует еще большое количество проблем:

- Отсутствует подходящий датасет, собранный для данной задачи

- С ростом количества вершин граф быстро становиться почти полносвязным, что усложняет его читаемость
- Нет общепринятого принципа маркировки вершин и в целом визуализации графа

Список литературы

[1] Customer support on twitter. `https://www.kaggle.com/datasets/thoughtvector/customer-support-on-twitter`. Доступ: 05.05.2022.

[2] Graphviz: open source graph visualization software. `https://graphviz.org/`. Доступ: 07.05.2022.

[3] Tadeusz Caliński and Harabasz JA. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3:1–27, 01 1974.

[4] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *CoRR*, abs/1506.02216, 2015.

[5] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[7] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. DailyDialog: A manually labelled multi-turn dialogue dataset. pages 986–995, November 2017.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[9] Apurba Nath and Aayush Kubba. TSCAN : Dialog structure discovery using SCAN. *CoRR*, abs/2107.06426, 2021.

[10] Liang Qiu, Yizhou Zhao, Weiyan Shi, Yuan Liang, Feng Shi, Tao Yuan, Zhou Yu, and Song-Chun Zhu. Structured attention for unsupervised dialogue structure induction. *CoRR*, abs/2009.08552, 2020.

[11] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. 11 2019.

[12] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[13] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[14] Weiyan Shi, Tiancheng Zhao, and Zhou Yu. Unsupervised dialog structure learning. *CoRR*, abs/1904.03736, 2019.

[15] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. 2020.

[16] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. 2020.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.

[18] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. 2020.

A Приложение

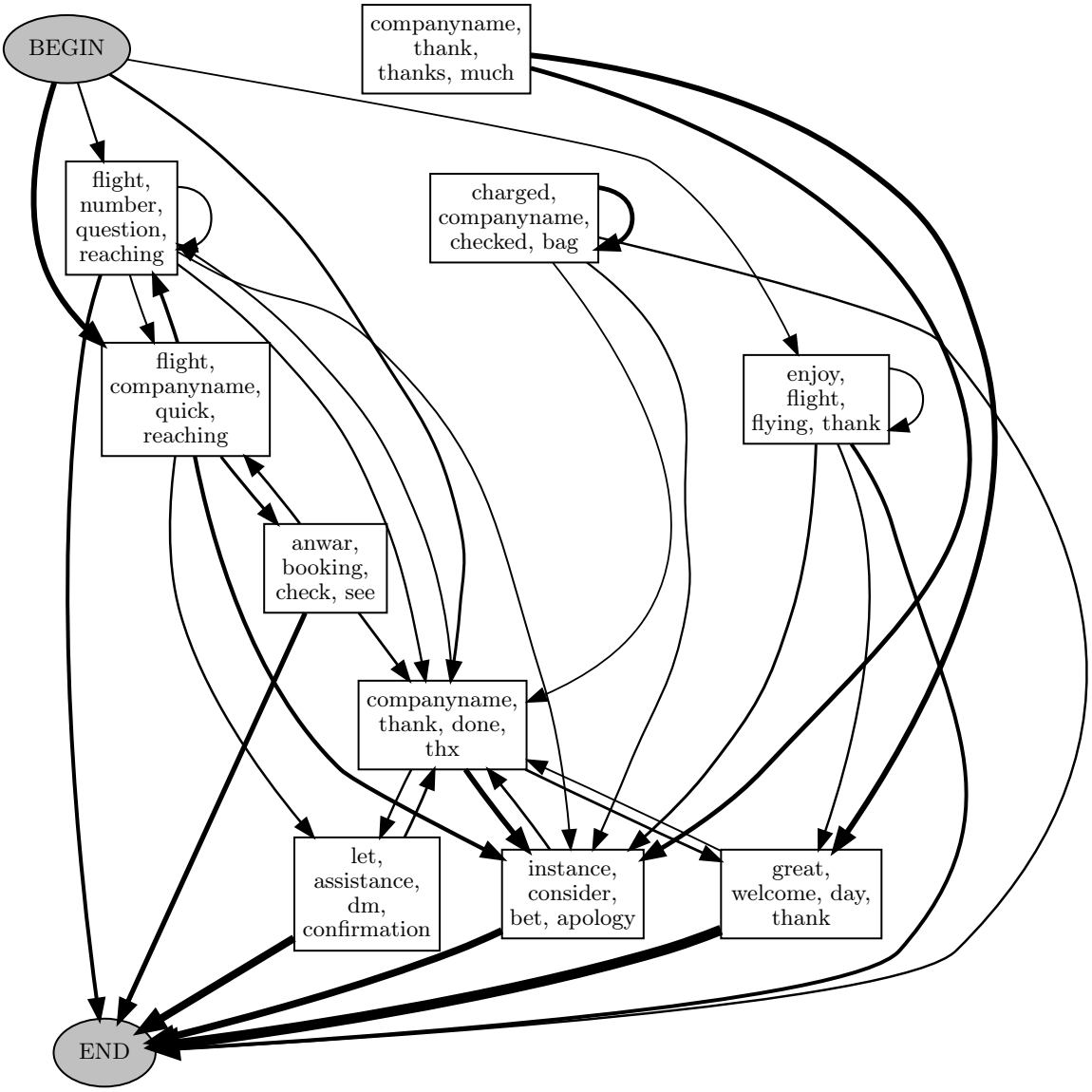


Рис. 8: Граф диалога с 10-ю вершинами для датасета Twitter Customer Support

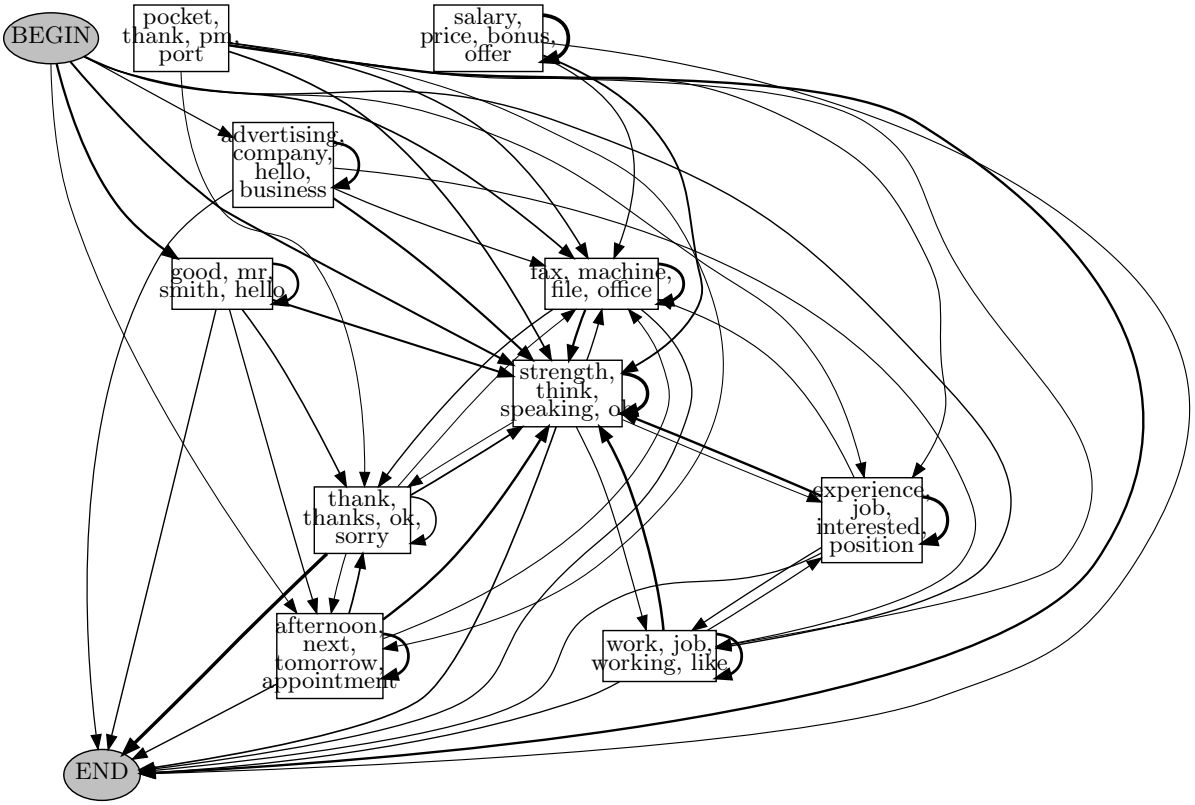


Рис. 9: Граф диалога с 10-ю вершинами для датасета DailyDialog

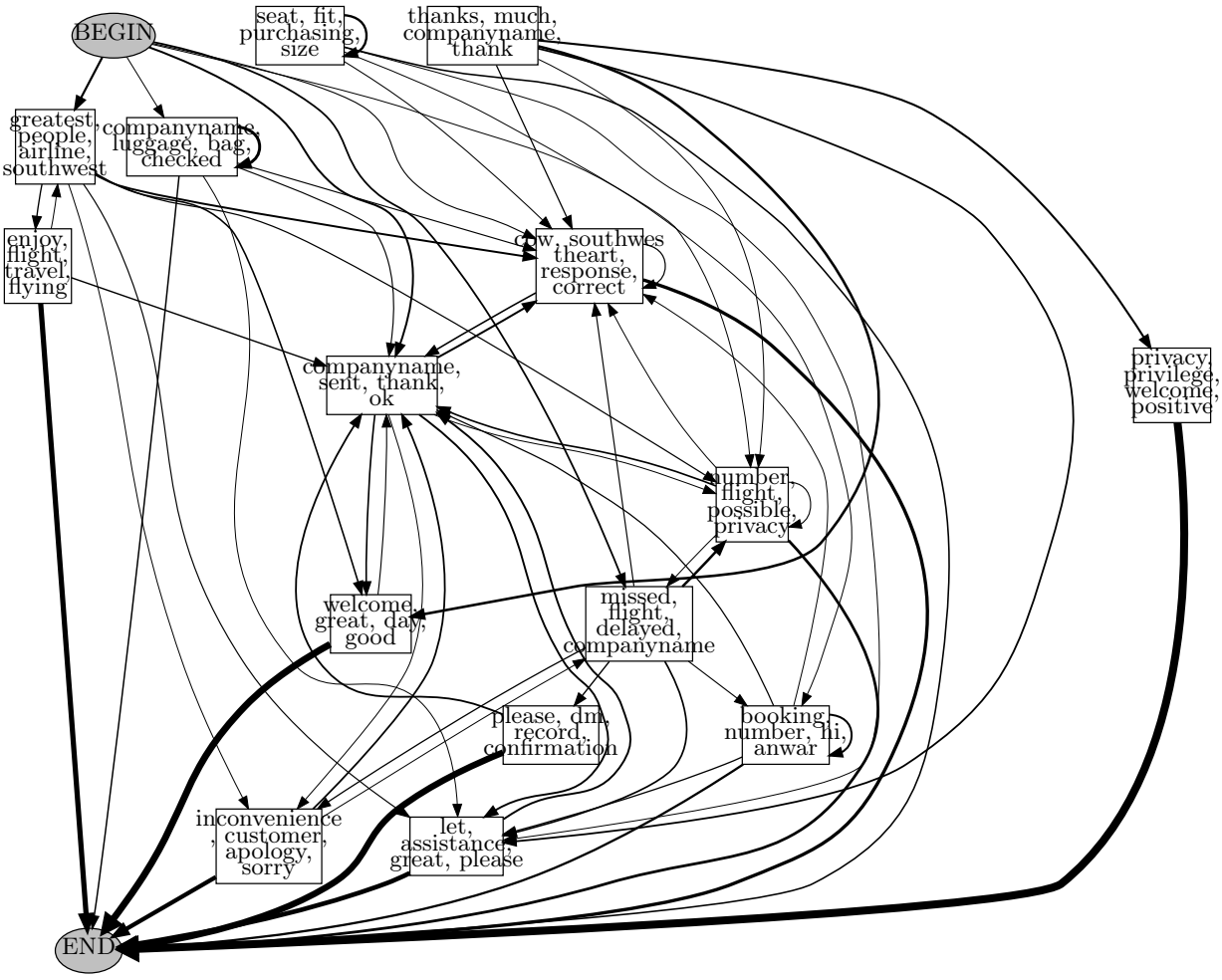


Рис. 10: Граф диалога с 15-ю вершинами для датасета Twitter Customer Support

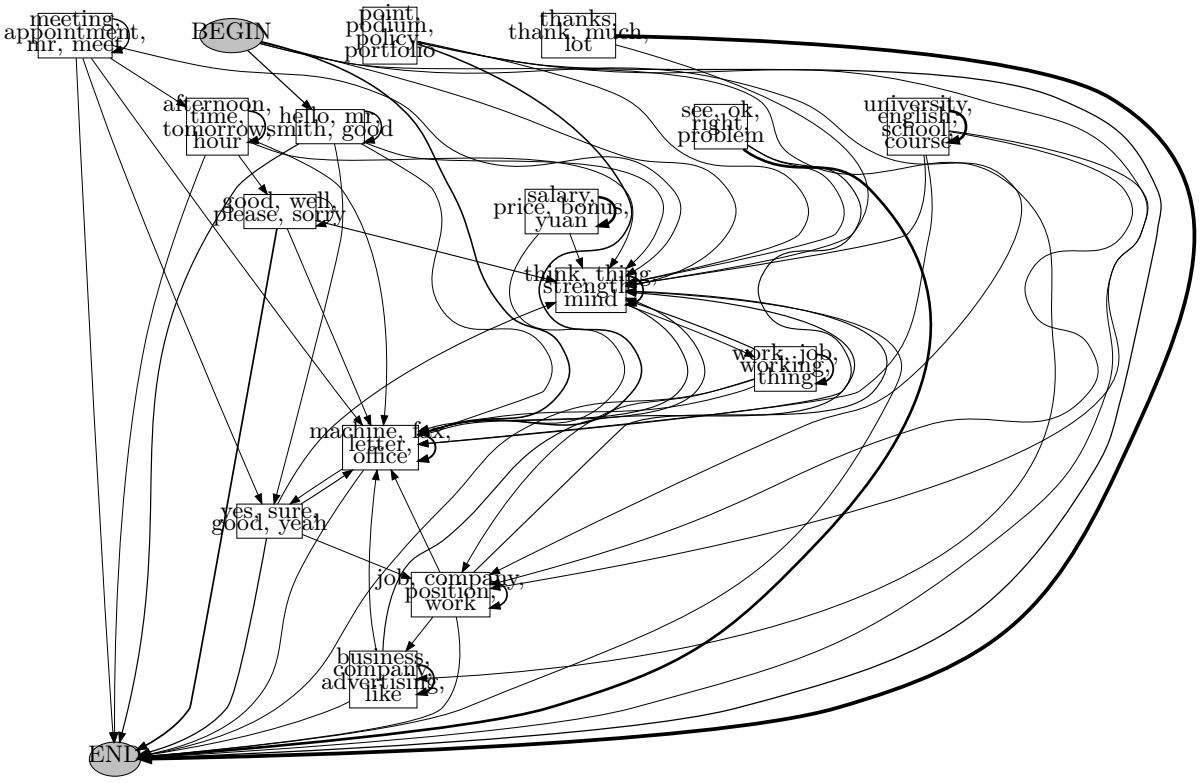


Рис. 11: Граф диалога с 10-ю вершинами для датасета DailyDialog