МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

УДК 519.7 DOI 10.17223/20710410/59/7

ПОСТРОЕНИЕ И ВИЗУАЛИЗАЦИЯ ОБОБЩЁННОГО ДИАЛОГОВОГО ГРАФА ПО КОРПУСУ ДИАЛОГОВ

П. Д. Штыков, А. Г. Дьяконов

Московский государственный университет имени М. В. Ломоносова, г. Москва, Россия

E-mail: shtykov.pa@gmail.com, djakonov@mail.ru

Предлагается определение обобщённого диалогового графа, с помощью которого описывается структура диалога по корпусу однородных диалогов. Задача построения такого графа является актуальной в современном разговорном искусственном интеллекте, однако работ с конкретными результатами мало, часто не даётся полного описания алгоритмов, не выкладывается код с их реализацией. В настоящей работе предложен метод построения обобщённого диалогового графа, который реализован на языке программирования Python и выложен в открытый доступ. Проведены эксперименты на открытых данных и описаны их результаты.

Ключевые слова: *диалоговая система, обработка естественного языка, граф, диалоговый граф, кластеризация, векторные представления.*

A GENERALIZED DIALOGUE GRAPH CONSTRUCTION AND VISUALIZATION BASED ON A CORPUS OF DIALOGUES

P. D. Shtykov, A. G. Dyakonov

Lomonosov Moscow State University, Moscow, Russia

A definition of a generalized dialogue graph is proposed to describe the structure of a dialogue in terms of a corpus of homogeneous dialogues. The task of constructing such a graph is relevant in modern conversational artificial intelligence, however, there are few works with meaningful results, often a complete description of the algorithms is not given and the code with the implementation is not published. In the paper, a method for constructing a generalized dialogue graph is proposed, which is implemented in the Python programming language and made publicly available. Experiments were carried out on open data and the results were described.

Keywords: *dialogue system, NLP, graph, dialogue graph, clustering, embeddings.*

Введение

Обработка естественного языка (Natural Language Processing, NLP) является одним из ключевых направлений в области искусственного интеллекта. При этом одной из важнейших задач в NLP является обработка и понимание диалогов. В данной работе

исследуется один из подходов к представлению и анализу диалогов, а также представлению общей структуры диалога в виде графа, идея которого не нова, но публикаций по его реализации крайне мало.

Предположим, что у нас есть достаточное число диалогов из некоторой узкой предметной области. Например, это диалоги работников колл-центра банка с клиентами (здесь диалоги ещё и проблемно-ориентированные — task-oriented dialogs). Естественно предположить, что их можно разбить на группы одноцелевых диалогов, например «предложение новой услуги», «перевыпуск банковской карты» и т. п. Каждой группе сопоставлен граф диалога; у каждого работника колл-центра есть чёткая инструкция по общению с клиентом и ей соответствует ориентированный граф (рис. 1).

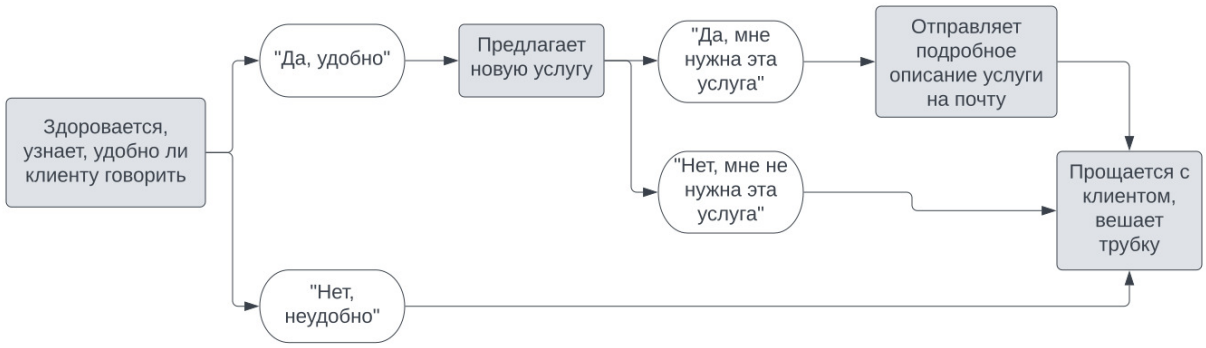


Рис. 1. Пример диалогового графа работника колл-центра с клиентом

По одному диалогу в общем случае невозможно восстановить диалоговый граф, поскольку не реализуются все варианты прохода по этому графу. По нескольким диалогам, которые соответствуют одному графу, это уже возможно. Теоретически (хороших практических реализаций этой идеи нет) можно восстановить набор графов по корпусу диалогов из некоторой узкой доменной области. Отметим специфику этой задачи:

— даже если графов несколько, у них есть общие вершины (например, вершина «поздороваться оператору»);
— не всегда диалог может идти по задуманному графу, возможны отклонения (например, пользователь просит повторить, отказывается от услуги и просит помощи в чём-то и т. п.);
— не всегда текущий ответ пользователя однозначно определяет переход на новую вершину графа (например, пользователь может попросить отключить услугу и закрыть счёт, оператор сам решает, с чего начать, и это определяет следующую вершину).

Отметим, что диалоговый граф, автоматически построенный по корпусу однородных диалогов, позволяет представить информацию в сжатой форме, подходящей как для визуализации, так и для встраивания в сложные диалоговые системы. В данной работе даётся понятие обобщённого диалогового графа и предлагаются способы его построения и визуализации (рассмотрим построение одного графа, а не набора).

1. Существующие подходы

Базовая идея построения диалогового графа состоит в поиске некоторой общей структуры диалогов в однородном корпусе. Такую структуру чаще всего представляют в виде ориентированного графа, вершины которого отражают темы реплик, а

дуги — переходы между ними (рис. 2). Приведём работы по изучению таких диалоговых графов.

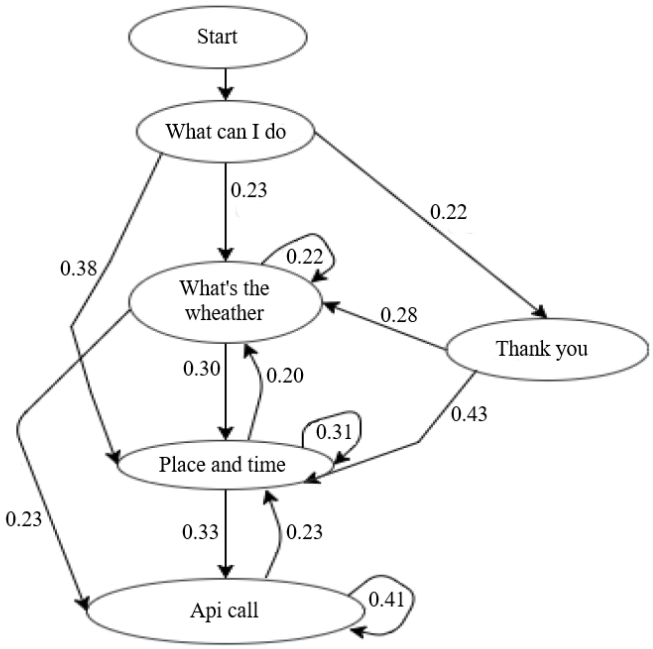


Рис. 2. Пример диалогового графа из [3]

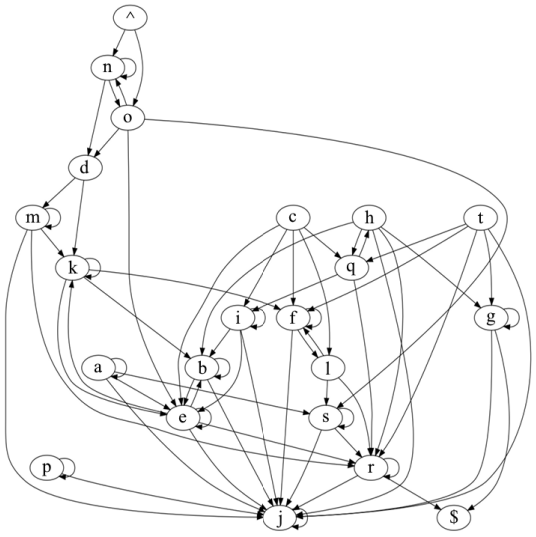
В [1, 2] предпринимались попытки обнаружения «структуры диалога», но без явного построения графа. Наиболее часто цитируемой является серия работ [3, 4], в которой предлагаются два способа построения такого графа. Первый основан на использовании рекуррентного вариационного автокодировщика [5] (Variational Recurrent Neural Network — VRNN), результат работы этого алгоритма представлен на рис. 2. Во второй авторы добавили в данную архитектуру механизм внимания [6], что позволило улучшить качество (Structured-Attention Variational Recurrent Neural Network — SVRNN).

В последние годы появляются работы на более амбициозные темы, например в [7] строится двухуровневый диалоговый граф для «открытого домена» (open domain dialogs). Обычно структура диалога выявляется для проблемно-ориентированных диалогов, здесь же рассматриваются более лексически разнообразные диалоги. В [7] используется довольно сложная техника: DVAE-GNN (Discrete Variational Auto-Encoder with Graph Neural Network).

На русском языке можно отметить работы [8, 9]. Первая выгодно отличается от многих наличием реализации в открытом доступе, в ней признаки, извлечённые из диалогового графа, используются при генерации реплик диалоговой системой.

В данной работе будем ориентироваться на метод TSCAN (Text SCAN) [10], в которой для построения диалогового графа применяется алгоритм классификации изображений с самообучением — SCAN (Semantic Clustering using Nearest Neighbors) [11]. Авторы адаптировали данный алгоритм для работы с текстами, использовав для получения векторных представлений (embeddings) текстов нейросеть BERT [12] архитектуры трансформер; один из примеров графа, полученного алгоритмом, приведён на рис. 3. В данной работе исследовано использование векторного представления, полученного с помощью модели SBERT (SentenceBERT, [13]), которое больше подходит

для семантической кластеризации. Однако в [10] авторы используют закрытый набор данных для сравнения алгоритма с более простыми методами, в частности с алгоритмом кластеризации k-средних (k-means) [14]. Кроме этого, авторы не предоставляют ни подробного алгоритма, ни его программного кода. В данной работе алгоритм реализован, выложен в открытый доступ, проведены эксперименты на открытых данных с разными методами кластеризации. Дополнено определение диалогового графа для более простого дальнейшего анализа определяемого объекта и визуализации.



Cluster-name	Self-label
^	Start Node
n	I am X
o	Am I talking to Y
d	I am calling from brand Z
m	This call is about product Z that you purchased last month
k	Have you made the payment for the last installment
f	The amount is five thousand three hundred dollars
l	Your due date is third of June
r	Your transaction id is five nine eight zero double two
g	When will the payment be done
$	End Node

Рис. 3. Пример диалогового графа из [10]

2. Постановка задачи

Пусть $D = \{d_1, d_2, \ldots, d_{|D|}\}$ — корпус диалогов, каждый диалог d_i является упорядоченным набором из нескольких высказываний: $d_i = \{d_i^1, d_i^2, \ldots, d_i^{n(i)}\}$. Каждое высказывание d_i^j — это текст, например «*К сожалению, такого товара нет в наличии. Можем ли мы Вам предложить что-то ещё*?». В дальнейшем мы будем работать с векторными представлениями текстов и высказывание d_i^j отождествлять с вещественным вектором. Множество всех высказываний в корпусе обозначим через

$$U = \bigcup_{i=1}^{|D|} \bigcup_{j=1}^{n(i)} d_i^j.$$

Мы будем работать с неразмеченными корпусами диалогов, в общем же случае нет ограничения на использование разметки (когда у каждого диалога или высказывания есть метка — некоторая дополнительная метаинформация). Для удобства считаем, что каждый диалог d_i начинается с «технического» высказывания «*начало диалога*»:

$$d_i^1 = \text{BEGIN},$$

а заканчивается «техническим» высказыванием «*конец диалога*»:

$$d_i^{n(i)} = \text{END},$$

аналогичные вершины будут и в диалоговом графе.

Определение 1. Назовём обобщённым диалоговым графом тройку

$$T = (G, p'(v'|v), p(u|v)),$$

где

— $G = (V, E)$ — ориентированный граф;

— $V \neq \varnothing$ — множество вершин графа G; $p(u|v)$ — функция вероятности отнесения высказывания $u \in U$ к вершине $v \in V$. Множество V интерпретируется как множество тем высказываний, поэтому $|V| < |U|$ (как правило, таких тем немного);

— E — множество дуг графа G, при этом каждой дуге $(v_j, v_i) \in E$ сопоставлена вероятность перехода по ней $p'(v_i|v_j)$, сумма вероятностей для дуг, выходящих из каждой вершины $v_j \in V$, равна 1: $\sum\limits_i p'(v_i|v_j) = 1$.

В определении есть произвол в выборе множеств V и E, формально в качестве G подойдёт любой ориентированный граф. Более того, если в качестве множества U взять не множество реплик диалогов корпуса, а множество всевозможных высказываний, то диалоговый граф не будет связан с конкретным корпусом. Однако понятна интерпретация указанных множеств: вершины $v \in V$ описывают темы реплик диалогов корпуса, а дуги $(v_j, v_i) \in E$ — чередование тем в диалогах. При этом функция p определяет связь между репликами и темами, а функция p' описывает смену тем. Основная задача при построении диалогового графа по корпусу диалога — чтобы описанные функции соответствовали чередованию реплик в диалогах конкретного корпуса. Дальше предлагается способ построения диалогового графа, в котором естественны указанные интерпретации.

Определение 1 не ограничивает нас в выборе модели для построения обобщённого диалогового графа. Обобщение (определений диалогового графа из предыдущих работ) связано с введением функций вероятности. Дополнительное требование наличия функции $p(u|v)$ позволяет вычислять статистики, полезные для визуализации и дальнейшего использования графа (например, самое вероятное предложение или самые частотные слова среди предложений, ассоциированных с текущей вершиной).

Такой граф достаточно просто обобщается на случай персонализированных диалогов (например, диалогов вида «пользователь» — «система») введением раскраски вершин, т. е. дополнительной функции $\phi(v)$, ставящей в соответствие каждой вершине персональный идентификатор пользователя (ID). Однако для корректности необходимо ввести дополнительные ограничения: смежные вершины не должны быть одинаково окрашены (так как высказывания пользователей чередуются), в графе нет петель (заметим, что основное определение их не запрещает). В данной работе мы будем строить диалоговый граф без дополнительной раскраски.

3. Предложенный метод

Чтобы не работать с текстами напрямую, в машинном обучении часто используют векторные представления: отображения вида

$$\mathrm{emb} : U \to \mathbb{R}^n, \quad n \in \mathbb{N},$$

которые реализуются с помощью нейронных сетей и позволяют работать не с текстом $u \in U$, а с вектором $\mathrm{emb}(u)$ фиксированной размерности. Для реализации представления мы использовали предобученную сиамскую нейронную сеть SBERT [13] с разными базовыми сетями (подробнее в п. 4). В дальнейшем, если не оговорено другого,

под высказыванием u будем подразумевать его представление $\mathrm{emb}(u)$ и отождествлять множество высказываний и множество их представлений, т. е.

$$U = \bigcup_{i=1}^{|D|} \bigcup_{j=1}^{n(i)} \mathrm{emb}(d_i^j) \subset \mathbb{R}^n.$$

При этом $n = 768$ или 384 в зависимости от использованной базовой сети в SBERT. В пространстве представлений введена косинусная мера сходства, отражающая семантическую близость высказываний, что позволяет использовать в этом пространстве простые методы кластеризации для объединения близких по смыслу высказываний.

Опишем алгоритм построения обобщённого диалогового графа $T{=}(G, p'(v'|v), p(u|v))$ для высказываний в пространстве представлений. Пусть выбран некоторый алгоритм кластеризации a, который разбивает множество U на кластеры — подмножества похожих высказываний. Множество полученных кластеров обозначим через V — оно будет множеством вершин графа. В результате работы алгоритма кластеризации определяется дискретная вероятность $p(v|u)$ принадлежности высказывания $u \in U$ к кластеру $v \in V$. При этом кластеризация может быть как жёсткой, например методом k-средних, так и мягкой, например смесью гауссиан (Gaussian mixture model, GMM) [14]. Зная $p(v|u)$, можно вычислить $p(u|v)$, используя теорему Байеса:

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

где $p(u)$ — вероятность встречаемости высказывания u во всем корпусе диалогов. На практике будем пользоваться оценками вероятностей — частотами. Заметим, что вероятность $p(u)$ не одинакова для всех высказываний, так как в диалогах корпуса могут встречаться повторяющиеся высказывания.

Отдельно потребуем, чтобы технические высказывания BEGIN и END попадали в отдельные кластеры {BEGIN} и {END}, для этих высказываний может не быть представлений, для удобства можно считать, что

$$\mathrm{emb}(\mathrm{BEGIN}) = (+\infty, \ldots, +\infty),$$
$$\mathrm{emb}(\mathrm{END}) = (-\infty, \ldots, -\infty)$$

(и их представления достаточно далеки от представлений других высказываний диалогов).

Осталось определить в графе G дуги и найти вероятности, ассоциированные с ними. Введём вспомогательный ориентированный граф $\hat{G} = (\hat{V}, \hat{E})$, множество вершин $\hat{V} = U$ — множество высказываний из корпуса, множество рёбер

$$\hat{E} = \bigcup_{i=1}^{|D|} \bigcup_{j=1}^{n(i)-1} \{(d_i^j, d_i^{j+1})\} \subset \hat{V} \times \hat{V}$$

— множество пар последовательных высказываний в диалогах корпуса, каждое ребро (u_i, u_j) имеет вероятностную метку $p''(u_j|u_i)$, равную апостериорной вероятности встретить ответ u_j на высказывание u_i. Данный граф строится напрямую по корпусу диалогов.

На рис. 4 показана схема совместного размещения обоих графов G и $\hat{G}$. Соответственно матрица смежности $\hat{A}$ графа $\hat{G}$ определяется как

$$\hat{A} = (\hat{a}_{ij}), \quad \hat{a}_{ij} = p''(u_j|u_i)$$

для $1 \leqslant i, j \leqslant |U|$. Зная вероятности $p(u|v)$, $p''(u|u)$ и $p(v|u)$, можно вычислить вероятности дуг $p'(v|v)$ в графе G:

$$p'(v_j|v_i) = \sum_{\alpha,\beta} p(v_j|u_\beta) p''(u_\beta|u_\alpha) p(u_\alpha|v_i),$$

т. е. вероятность перехода из вершины v_i в вершину v_j графа G равна сумме вероятностей всех простых путей из v_i в v_j, проходящих через пары высказываний вида (u_α, u_β) в графе $\hat{G}$. Пример такого пути выделен на рис. 4 штрихпунктирной линией.

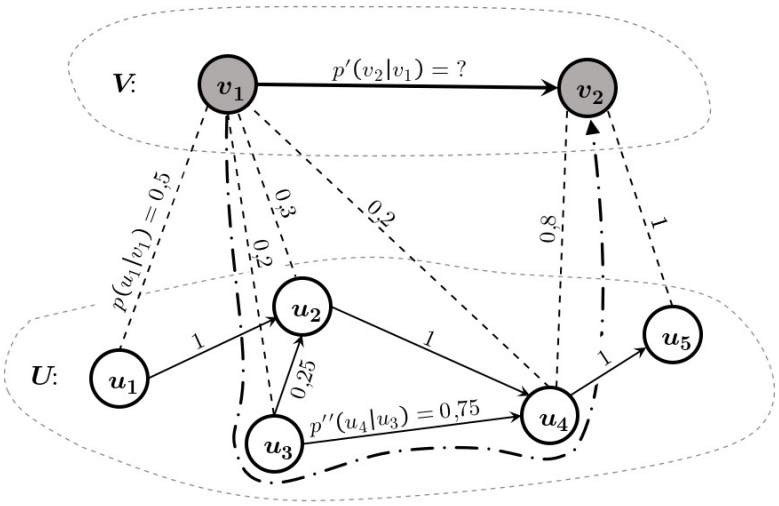


Рис. 4. Пример двух графов: G (сверху) в пространстве вершин V и $\hat{G}$ (снизу) в пространстве высказываний U

Вероятностная матрица смежности взвешенного графа G имеет вид

$$A = (a_{ij}), \quad a_{ij} = p'(v_j|v_i)$$

для $1 \leqslant i, j \leqslant |V|$. Так как в нашем случае совместные распределения $p(u|v)$ и $p(v|u)$ дискретны, то они могут быть представлены в виде матриц, поэтому способ вычисления матрицы смежности A может быть задан в явной матричной форме

$$A = P^{uv} \cdot \hat{A} \cdot P^{vu},$$

где матрицы в произведении определяются следующим образом:

$$P^{uv} = (p_{ij}) : \; p_{ij} = p(u_j|v_i), \quad P^{vu} = (p_{ij}) : \; p_{ij} = p(v_j|u_i).$$

Описание построения обобщённого диалогового графа T закончено. Заметим, что этод метод применим не только в случае кластеризации в пространстве представлений, но и в случае использования любого другого алгоритма, способного оценить апостериорные вероятности $p(v|u)$ (например, с помощью латентного размещения Дирихле (Latent Dirichlet allocation, LDA [15]) или нейронной сети, решающей задачу «от начала до конца» без промежуточного использования представлений). Кластеризация была выбрана как наиболее простой метод. Исходный код алгоритма и экспериментов доступен по ссылке [16].

4. Эксперименты

4.1. Данные для экспериментов

Для многих задач в анализе данных и машинном обучении есть стандартные наборы данных (так называемые «датасеты» — datasets), на которых отслеживается качество предложенных решений и определяется лучшее текущее решение — SotA (state of the Art), например на ресурсе `paperswithcode.com`. Постановка задачи, рассматриваемая в данной работе, достаточно нова: стандартных наборов данных, т. е. корпусов диалогов, имеющих некоторую общую *известную* структуру, почти нет. Этому критерию удовлетворяет лишь корпус STAR (Schema-Guided Dialog Dataset for Transfer Learning) [17], на котором не тестировались упомянутые выше методы. Поэтому для проведения экспериментов выбраны два известных корпуса, для которых можно предположить наличие общей структуры. Первый корпус — Customer Support on Twitter [18], в котором собраны ответы официальных аккаунтов технической поддержки крупных компаний. Для обеспечения однородности из него выбрано подмножество сообщений аккаунтов шести разных авиакомпаний США. Второй корпус — DailyDialog [19], в котором собраны обычные диалоги из повседневной жизни на разные темы. Для экспериментов были взяты диалоги на тему работы, как наиболее однородные. В результате подготовлены два набора данных: 8081 диалог в среднем по 3,6 высказываний в диалоге и 1924 диалога в среднем по 7,5 высказываний. Примеры диалогов из двух наборов приведены на рис. 5.

Twitter Customer Support

— @AlaskaAir it says you open at 5:15 @317258 where is everyone? #hellooooooo 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. 5. Примеры диалогов из корпусов Twitter Customer Support и DailyDialog

Видно, что корпус Twitter Customer Support очень зашумлён, так как переписка в соцсети Twitter публична и в неё часто вмешиваются третьи лица. Поэтому в корпусе были оставлены только те диалоги между компаниями и пользователями, которые соответствуют следующей схеме:

«система» — «пользователь N» — «система» — «пользователь N» и т. д.

Из высказываний были убраны все идентификаторы пользователей, а идентификаторы компаний были заменены на единый токен «companyname». После этого была применена следующая предобработка текста:

— приведение всего текста к нижнему регистру;
— удаление слов с цифрами;

— удаление ссылок;
— лемматизация с помощью пакета NLTK [20];
— удаление стоп-слов (использовался набор стоп-слов из пакета NLTK).

Корпус DailyDialog зашумлён существенно меньше, поэтому к нему была применена только описанная предобработка.

4.2. Оценка качества кластеризации

Сначала рассмотрим визуализацию пространства представлений, полученных с помощью SBERT с базовой нейронной сетью Distill-RoBERTa [21]. Пространство отображено на плоскость с помощью t-SNE (t-distributed Stochastic Neighbor Embedding) [22] с перплексией равной 50 (рис. 6 и 7). Для диалогов из обоих корпусов заметна кластерная структура, однако кластеры имеют небольшие размеры и между ними много шума. Это может привести к зашумлению и самого диалогового графа.

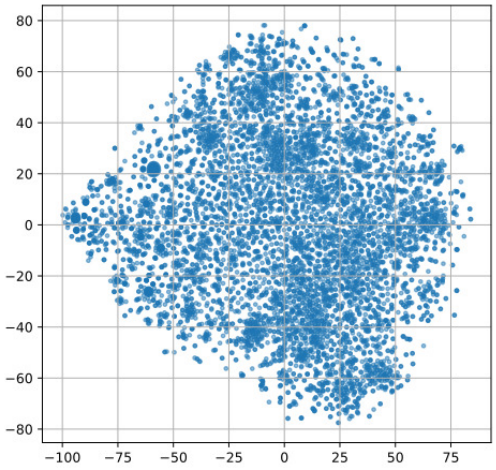


Рис. 6. Пространство представлений для корпуса DailyDialog

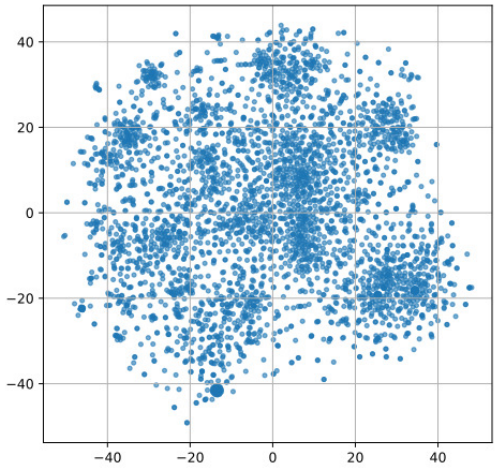


Рис. 7. Пространство представлений для корпуса Twitter Customer Support

Измерим качество кластеризации. В табл. 1 представлены результаты работы алгоритма с 5-ю вершинами в графе в зависимости от следующих параметров:

— базовая модель в SBERT: MPNet [23], DistillRoBERTa [21, 24] и MiniLM [25];
— кластеризатор: k-средних, смесь гауссиан (GMM) и SCAN [10].

Так как в [10] не указаны оптимальные гиперпараметры для SCAN, он обучался для каждой конфигурации с нуля со следующими стандартными гиперпараметрами:

— количество голов классификатора: 1;
— темп обучения: 10^{-5};
— количество эпох: 15.

Для графов с количеством вершин 10 и 15 аналогичная статистика приведена в табл. 2 и 3.

Таблица 1

Результаты сравнения качества кластеризации для графа с 5-ю вершинами для исследованных наборов данных

	Model	Twitter Customer Support				DailyDialog			
		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**
	MPNet_SCAN	0,042	**1117,5**	4,169	0,576	0,017	230,8	5,992	0,717
	RoBERTa_SCAN	0,037	947,8	4,569	0,625	0,024	258,9	5,563	0,718
	MiniLM_SCAN	0,036	901,7	4,493	0,624	0,021	247,2	5,596	0,705

Таблица 2

Результаты сравнения качества кластеризации для графа с 10-ю вершинами для корпусов Twitter Customer Support и DailyDialog

	Model	Twitter Customer Support				DailyDialog			
		Silh.	C.-H.	D.-B.	Entr.	Silh.	C.-H.	D.-B.	Entr.
$\|V\| = 10$	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**
	MPNet_SCAN	0,032	626,7	4,569	0,663	0,014	176,7	5,117	0,809
	RoBERTa_SCAN	0,031	563,0	4,465	0,68	0,021	165,6	5,414	0,809
	MiniLM_SCAN	0,03	585,5	4,278	0,679	0,021	175,4	5,003	0,81

Таблица 3

Результаты сравнения качества кластеризации для графа с 15-ю вершинами для корпусов Twitter Customer Support и DailyDialog

	Model	Twitter Customer Support				DailyDialog			
		Silh.	C.-H.	D.-B.	Entr.	Silh.	C.-H.	D.-B.	Entr.
$\|V\| = 15$	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
	MPNet_SCAN	0,024	475,5	4,457	0,693	0,005	136,3	4,779	0,832
	RoBERTa_SCAN	0,024	447,4	4,418	0,694	0,023	135,2	4,937	0,829
	MiniLM_SCAN	0,024	439,9	4,43	0,696	0,022	140,1	5,132	0,826

В качестве показателей качества кластеризации использовались следующие базовые для неразмеченных данных: коэффициент силуэта (Silh.) [26], индекс Калински — Харабаса (C.-H.) [27] и индекс Дэвиса — Болдина (D.-B.) [28]. Введём дополнительный показатель для оценки структуры графа. Нам хотелось бы, чтобы граф был «более детерминированным»: если случайно блуждать по графу согласно приписанным дугам вероятностям, то в идеале переходы должны быть детерминированы (т. е. только

одна исходящая из вершины дуга имеет вероятность 1, а остальные — 0). Для этого будем измерять среднюю нормализованную энтропию (Entr.):

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

Чем меньше энтропия, тем более детерминирован граф.

Из табл. 1–3 видно, что мы не смогли в точности повторить результаты авторов TSCAN [10] их же методом: наша реализация SCAN-кластеризатора уступает стандартным алгоритмам k-средних и смеси гауссиан по всем показателям. Возможно, это связано с неправильно подобранными гиперпараметрами.

4.3. Визуализация графов

Построим и визуализируем графы, полученные предложенным методом. Для всех графов использовалась лучшая (по результатам из табл. 1–3) модель для данного количества вершин и набора данных. В качестве маркировки вершин будем использовать четыре слова из высказываний, которые соответствуют вершине, с самым большим значением Tf-Idf (TF — term frequency, IDF — inverse document frequency) [29]. Tf-Idf-представления строились для двухсот наиболее вероятных для данной вершины высказываний. Для удобства визуализации дуги не помечаются вероятностями, вероятность отображается толщиной дуги: чем толще дуга, тем больше вероятность перехода по ней. Также убраны дуги с вероятностями меньше 0,1. Визуализация графов производилась с помощью пакета GraphViz [30].

На рис. 8 и 9 представлены графы с 5-ю вершинами, составленные по обоим корпусам, на рис. 10 и 11 — с 10-ю вершинами, на рис. 12 и 13 — графы с 15-ю вершинами.

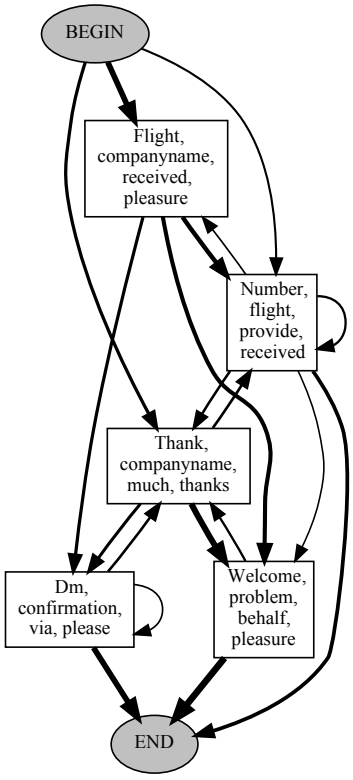


Рис. 8. Диалоговый граф с 5-ю вершинами для корпуса Twitter Customer Support

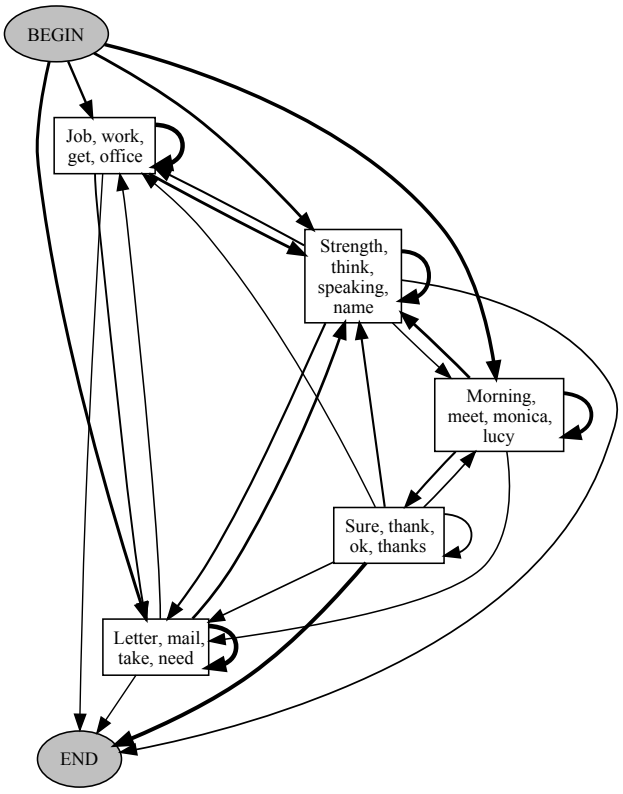


Рис. 9. Диалоговый граф с 5-ю вершинами для корпуса DailyDialog

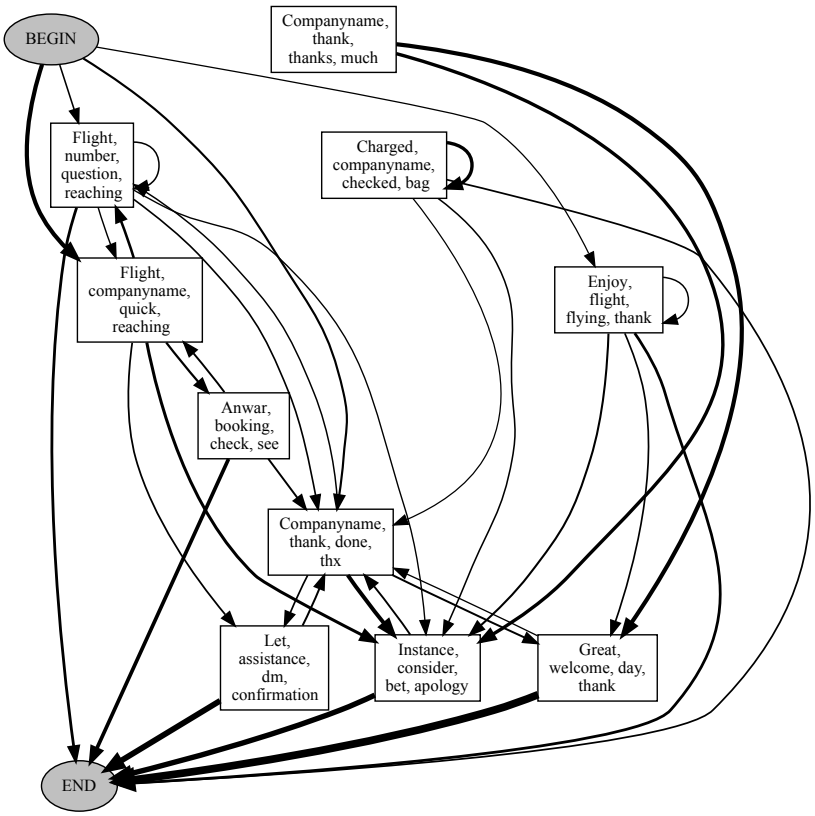


Рис. 10. Диалоговый граф с 10-ю вершинами для корпуса Twitter Customer Support

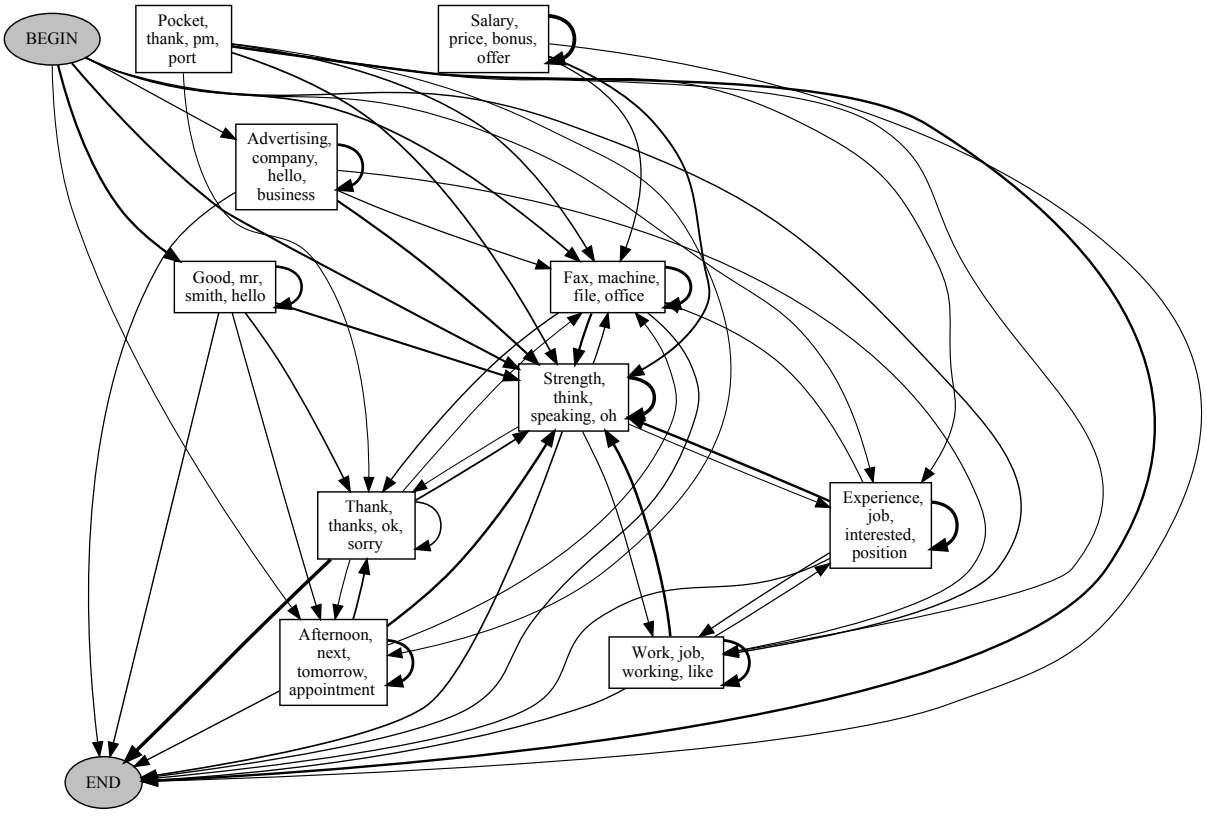


Рис. 11. Диалоговый граф с 10-ю вершинами для корпуса DailyDialog

Графы с 5-ю вершинами выглядят приемлемыми для анализа, дуги с разной толщиной позволяют понять, какие диалоги наиболее вероятны. Хотя Tf-Idf-представление является простым инструментом маркировки вершин, оно позволяет понять тему высказываний, которые соответствуют вершине. Графы с 10-ю и 15-ю вершинами становятся почти полносвязными, наиболее вероятные диалоги на них менее заметны, интерпретировать такие графы сложнее.

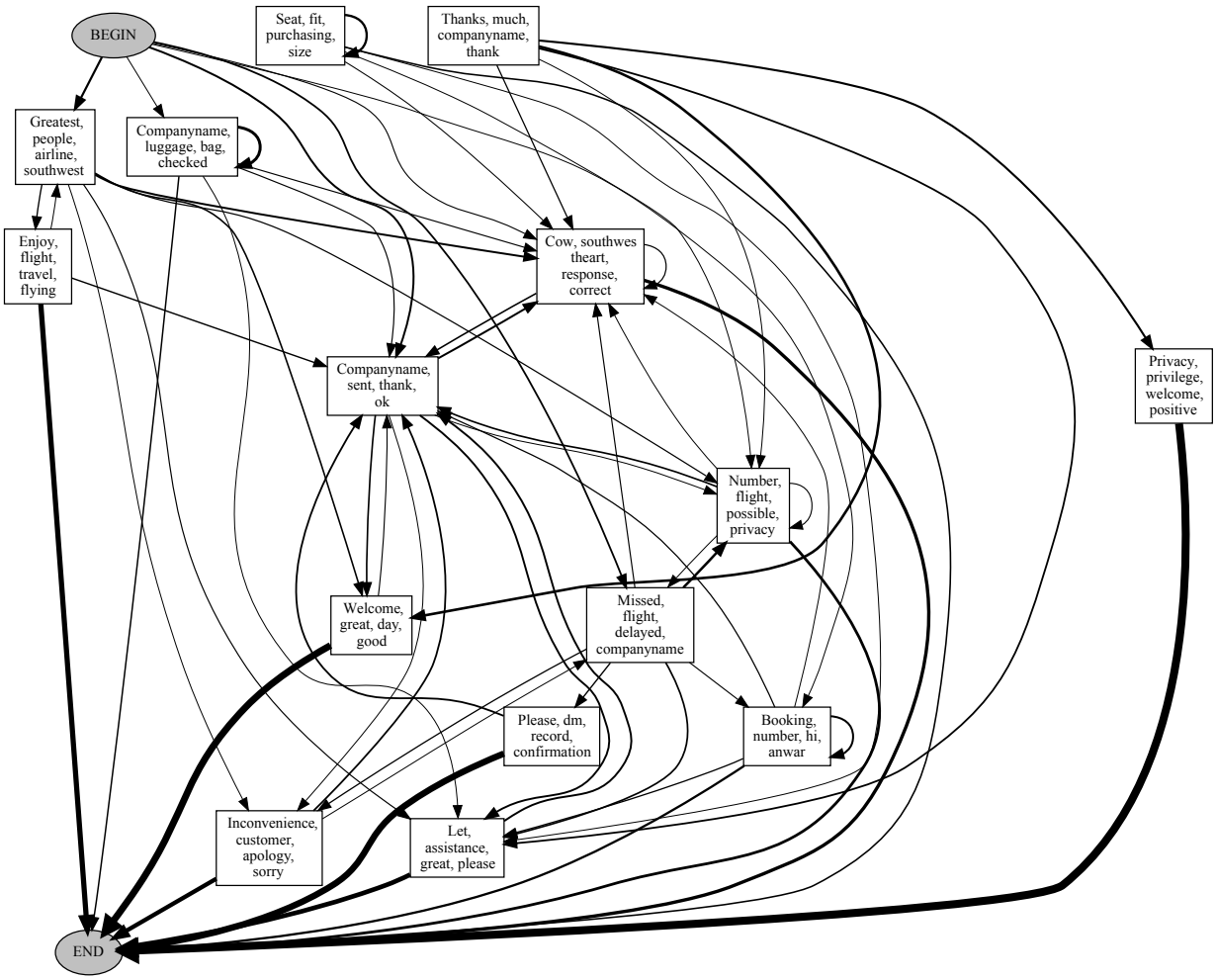


Рис. 12. Диалоговый граф с 15-ю вершинами для корпуса Twitter Customer Support

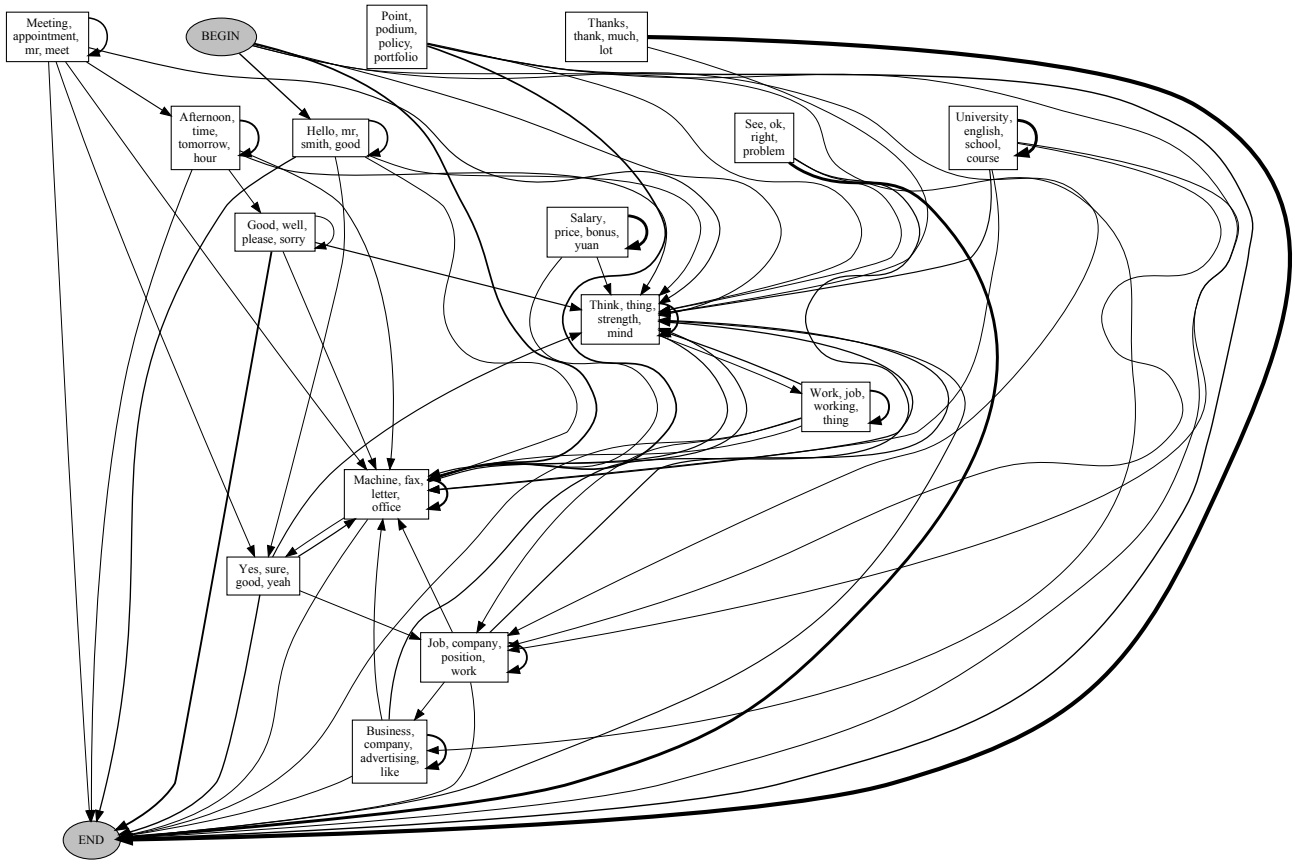


Рис. 13. Диалоговый граф с 15-ю вершинами для корпуса DailyDialog

Заключение

Предложен и исследован на двух наборах данных алгоритм построения обобщённого диалогового графа с помощью кластеризации в пространстве представлений SBERT. Проведены сравнения простых методов кластеризации со SCAN. В результате получены изображения диалоговых графов, пригодные для визуального анализа. Отметим, что работа оставляет довольно большой задел для будущих исследований:

— автоматическое определение числа вершин в графе (простейший вариант решения — использование алгоритма DBSCAN, см., например, [8, 9]);
— автоматическое определение высказываний, которые не соответствуют вершинам (например, отклонения от темы, здесь также можно использовать DBSCAN);
— автоматическая пометка вершин (хотелось бы, чтобы оно производилось полноценным предложением, здесь напрашивается применить технику реферирования, в [8] рассматривался вариант использования высказывания, чьё представление наиболее близко к центру соответствующего кластера);
— исследование оптимального представления высказываний и оптимальной кластеризации (это более эффективно решается с наборами данных, заточенных под решаемую задачу, например, если диалоговые графы построены экспертами или заданы изначально как в STAR [17]);
— проблема сравнения диалоговых графов (в идеале сравниваются ответы разных алгоритмов — графы, а не промежуточные результаты их работы — кластеризации);
— проблема построения нескольких графов (как описано во введении, по-видимому, такая постановка задачи ранее не рассматривалась).

Авторы выражают благодарность анонимному рецензенту за внимание к работе и полезные замечания, которые помогли существенно улучшить статью.

ЛИТЕРАТУРА

1. *Chotimongkol A.* Learning the structure of task-oriented conversations from the corpus of in-domain dialogs. PhD thesis. Carnegie Mellon University, 2008.
2. *Tang D., Li X., Gao J., et al.* Subgoal discovery for hierarchical dialogue policy learning // Proc. EMNLP. Brussels, Belgium, 2018. P. 2298–2309.
3. *Shi W., Zhao T., and Yu Z.* Unsupervised Dialog Structure Learning. ArXiv. 2019. arxiv.org/abs/1904.03736.
4. *Qiu L., Zhao Y., Shi W., et al.* Structured Attention for Unsupervised Dialogue Structure Induction. ArXiv. 2020. arxiv.org/abs/2009.08552.
5. *Chung J., Kastner K., Dinh L., et al.* A Recurrent Latent Variable Model for Sequential Data. ArXiv. 2015. arxiv.org/abs/1506.02216.
6. *Vaswani A., Shazeer N., Parmar N., et al.* Attention Is All You Need. ArXiv. 2017. arxiv.org/abs/1706.03762.
7. *Xu J., Lei Z., Wang H., et al.* Discovering Dialog Structure Graph for Open-Domain Dialog Generation. ArXiv. 2020. arxiv.org/abs/2012.15543.
8. *Юсупов И. Ф., Трофимова М. В., Бурцев М. С.* Построение и использование диалогового графа для улучшения оценки качества в целенаправленном диалоге // ТРУДЫ МФТИ. 2020. Т. 21. № 3. С. 75–86.
9. *Фельдина Е. А., Махныткина О. В.* Автоматическое построение дерева диалога по неразмеченным текстовым корпусам на русском языке // Научно-технический вестник информационных технологий, механики и оптики. 2021. Т. 21. № 5. С. 709–719.
10. *Nath A. and Kubba A.* TSCAN: Dialog Structure Discovery using SCAN. ArXiv. 2021. arxiv.org/abs/2107.06426.
11. *Van Gansbeke W., Vandenhende S., Georgoulis S., et al.* SCAN: Learning to Classify Images without Labels. ArXiv. 2020. arxiv.org/abs/2005.12320.
12. *Devlin J., Chang M.-W., Lee K., and Toutanova K.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv. 2018. arxiv.org/abs/1810.04805.
13. *Reimers N. and Gurevych I.* Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks // ArXiv. 2019. arxiv.org/abs/1908.10084.
14. *Bishop C.* Pattern Recognition and Machine Learning. N.Y.: Springer, 2006. 738 p.
15. *Blei D., Ng A., and Jordan M.* Latent Dirichlet allocation // J. Machine Learning Res. 2003. V. 3. P. 993–1022.
16. https://github.com/PavelShtykov/generalized_dialogue_graph — Построение и визуализация обобщённого диалогового графа по корпусу диалогов. 2022.
17. *Mosig J., Mehri S., and Kober T.* STAR: A Schema-Guided Dialog Dataset for Transfer Learning. ArXiv. 2020. arxiv.org/abs/2010.11853.
18. www.kaggle.com/datasets/thoughtvector/customer-support-on-twitter — Customer Support on Twitter. 2022.
19. *Li Y., Su H., Shen X., et al.* DailyDialog: A manually labelled multi-turn dialogue dataset // Proc. Eighth Int. Joint Conf. Natural Language Processing. Taipei, Taiwan, 2017. V. 1. P. 986–995.
20. https://www.nltk.org — Natural Language Toolkit. 2022.
21. *Liu Y., Ott M., Goyal N., et al.* RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv. 2019. arxiv.org/abs/1907.11692.

22. *Van der Maaten L. and Hinton G.* Viualizing data using t-SNE // J. Machine Learning Res. 2008. V. 9. P. 2279–2605.
23. *Song K., Tan X., Qin T., et al.* MPNet: Masked and Permuted Pre-training for Language Understanding. ArXiv. 2020. arxiv.org/abs/2004.09297.
24. *Sanh V., Debut L., Chaumond J., and Wolf T.* DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. ArXiv. 2019. arxiv.org/abs/1910.01108.
25. *Wang W., Wei F., Dong L., et al.* MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. ArXiv. 2020. arxiv.org/abs/2002.10957.
26. *Rousseeuw P. J.* Silhouettes: A graphical aid to the interpretation and validation of cluster analysis // J. Comput. Appl. Math. 1987. V. 20. P. 53–65.
27. *Caliński T. and Harabasz J.* A dendrite method for cluster analysis // Commun. in Statistics — Theory and Methods. 1974. V. 3. No. 1. P. 1–27.
28. *Davies D. L. and Bouldin D. W.* A cluster separation measure // IEEE Trans. Pattern Analysis and Machine Intelligence. 1979. V. 1. No. 2. P. 224–227.
29. *Spärck K. J.* A statistical interpretation of term specificity and its application in retrieval // J. Documentatio. 2004. V. 60. P. 493–502.
30. https://graphviz.org — Graphviz: open source graph visualization software. 2022.

REFERENCES

1. *Chotimongkol A.* Learning the structure of task-oriented conversations from the corpus of in-domain dialogs. PhD thesis, Carnegie Mellon University, 2008.
2. *Tang D., Li X., Gao J., et al.* Subgoal discovery for hierarchical dialogue policy learning. Proc. EMNLP, Brussels, Belgium, 2018, pp. 2298–2309.
3. *Shi W., Zhao T., and Yu Z.* Unsupervised Dialog Structure Learning. ArXiv, 2019, arxiv.org/abs/1904.03736.
4. *Qiu L., Zhao Y., Shi W., et al.* Structured Attention for Unsupervised Dialogue Structure Induction. ArXiv, 2020, arxiv.org/abs/2009.08552.
5. *Chung J., Kastner K., Dinh L., et al.* A Recurrent Latent Variable Model for Sequential Data. ArXiv, 2015, arxiv.org/abs/1506.02216.
6. *Vaswani A., Shazeer N., Parmar N., et al.* Attention Is All You Need. ArXiv, 2017, arxiv.org/abs/1706.03762.
7. *Xu J., Lei Z., Wang H., et al.* Discovering Dialog Structure Graph for Open-Domain Dialog Generation. ArXiv, 2020, arxiv.org/abs/2012.15543.
8. *Yusupov I. F., Trofimova M. V., and Burtsev M. S.* Postroenie i ispol'zovanie dialogovogo grafa dlya uluchsheniya otsenki kachestva v tselenapravlennom dialoge [Unsupervised graph extraction for improvement of multi-domain task-oriented dialogue modelling]. Proc. MIPT, 2020, vol. 21, no. 3, pp. 75–86. (in Russian)
9. *Fel'dina E. A. and Makhnytkina O. V.* Avtomaticheskoe postroenie dereva dialoga po nerazmechennym tekstovym korpusam na russkom yazyke [Automatic construction of the dialog tree based on unmarked text corpora in Russian]. Nauchno-Tekhnicheskiy Vestnik Informatsionnykh Tekhnologiy, Mekhaniki i Optiki, 2021, vol. 21, no 5, pp. 709–719. (in Russian)
10. *Nath A. and Kubba A.* TSCAN: Dialog Structure Discovery using SCAN. ArXiv, 2021, arxiv.org/abs/2107.06426.
11. *Van Gansbeke W., Vandenhende S., Georgoulis S., et al.* SCAN: Learning to Classify Images without Labels. ArXiv, 2020, arxiv.org/abs/2005.12320.
12. *Devlin J., Chang M.-W., Lee K., and Toutanova K.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv, 2018, arxiv.org/abs/1810.04805.

13. *Reimers N. and Gurevych I.* Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks // ArXiv, 2019, arxiv.org/abs/1908.10084.
14. *Bishop C.* Pattern Recognition and Machine Learning. N.Y., Springer, 2006. 738 p.
15. *Blei D., Ng A., and Jordan M.* Latent Dirichlet allocation. J. Machine Learning Res., 2003, vol. 3, pp. 993–1022.
16. https://github.com/PavelShtykov/generalized_dialogue_graph. 2022.
17. *Mosig J., Mehri S., and Kober T.* STAR: A Schema-Guided Dialog Dataset for Transfer Learning. ArXiv, 2020, arxiv.org/abs/2010.11853.
18. www.kaggle.com/datasets/thoughtvector/customer-support-on-twitter — Customer Support on Twitter, 2022.
19. *Li Y., Su H., Shen X., et al.* DailyDialog: A manually labelled multi-turn dialogue dataset. Proc. Eighth Int. Joint Conf. Natural Language Processing, Taipei, Taiwan, 2017. vol. 1, pp. 986–995.
20. https://www.nltk.org — Natural Language Toolkit, 2022.
21. *Liu Y., Ott M., Goyal N., et al.* RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv, 2019, arxiv.org/abs/1907.11692.
22. *Van der Maaten L. and Hinton G.* Viualizing data using t-SNE. J. Machine Learning Res., 2008, vol. 9, pp. 2279–2605.
23. *Song K., Tan X., Qin T., et al.* MPNet: Masked and Permuted Pre-training for Language Understanding. ArXiv, 2020, arxiv.org/abs/2004.09297.
24. *Sanh V., Debut L., Chaumond J., and Wolf T.* DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. ArXiv, 2019, arxiv.org/abs/1910.01108.
25. *Wang W., Wei F., Dong L., et al.* MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. ArXiv, 2020, arxiv.org/abs/2002.10957.
26. *Rousseeuw P. J.* Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math., 1987, vol. 20, pp. 53–65.
27. *Caliński T. and Harabasz J.* A dendrite method for cluster analysis. Commun. in Statistics — Theory and Methods, 1974, vol. 3, no. 1, pp. 1–27.
28. *Davies D. L. and Bouldin D. W.* A cluster separation measure. IEEE Trans. Pattern Analysis and Machine Intelligence, 1979, vol. 1, no. 2, pp. 224–227.
29. *Spärck K. J.* A statistical interpretation of term specificity and its application in retrieval. J. Documentatio, 2004, vol. 60, pp. 493–502.
30. https://graphviz.org — Graphviz: open source graph visualization software, 2022.