Сколковский институт науки и технологий

На правах рукописи

Александр Иванович Панченко

Методы и алгоритмы для извлечения, связывания, векторизации и разрешения неоднозначности лексико-семантических графов

Резюме диссертации
на соискание ученой степени
доктора компьютерных наук

Москва – 2024

Данная докторская диссертация была подготовлена в Сколковском институте науки и технологий (Сколтех) и частично в Исследовательском институте искусственного интеллекта (AIRI) на основе публикаций, подготовленных Гамбургском университете, Техническом университете Дармштадта, Сколтехе и AIRI. Самая ранняя публикация, связанная с этой диссертацией, была опубликована в 2016 годом, в то время как, последняя была опубликована в 2023 году. Ниже представлено краткое изложение основных методов, предложенных в рамках диссертационного исследования.

Оглавление

Глава 1

Введение

Лексические ресурсы, такие как Wordnet[1], таксономии, тезаурусы и словари, содержат точную закодированную вручную информацию о словах, словосочетаниях и отношениях между ними, таких как синонимия или гипернимия. Однако охват и актуальность подобных ресурсов часто ограничены. Это связано с дорогостоящим, длительным и, как правило, полностью ручным процессом создания ресурса и его поддержки в актуальном состоянии, требующим труда лексикографов, лингвистов и экспертов предметной области. Кроме того, некоторые термины, специфичные для той или иной области, могут отсутствовать даже в самых больших лексикографических ресурсах, таких как Викисловарь.[2]

С другой стороны, активно разрабатываются подходы, основанные на данных, такие как дистрибутивные семантические модели и методы извлечения информации, для поиска значений слов и отношений между ними из больших текстовых корпусов, таких как Википедия[3] или CommonCrawl[4]. Этот альтернативный автоматический способ построения лексико-семантических ресурсов, в отличие от ручного подхода, обычно дает высокую полноту из-за огромного лексического покрытия неразмеченных текстовых корпусов, но его

[1] `https://wordnet.princeton.edu`
[2] `https://www.wiktionary.org`
[3] `https://www.wikipedia.org`
[4] `https://commoncrawl.org`

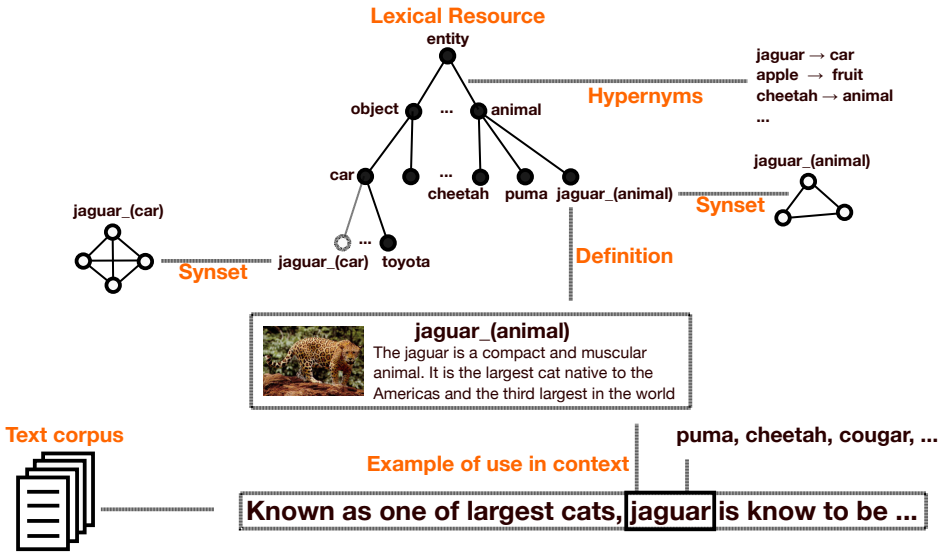

Рис. 1-1: Обзор основных понятий вычислительной лексической семантики и взаимосвязей между ними, использованных в диссертации.

результаты работы подобных методов могут иметь низкую точность. Одной из важнейших целей данной работы было преодоление разрыва между этими двумя подходами к вычислительной лексической семантике: использовать лучшее из обоих подходов и достичь высокой точности ручных ресурсов при сохранении высокой полноты автоматических методов.

Рис. 1-1 иллюстрирует основные лингвистические концепции и взаимодействия между ними, которые моделируются и обрабатываются с использованием вычислительных методов, предложенных в этой диссертации. **Лексический ресурс** представляет собой графа $G = (V, E)$ с набором узлов V, представляющим **concepts**, и набором ребер E, представляющим **семантические отношения** между ними, такие как **синонимия**, например $e_i = (car, vehicle)$ или **гиперонимия**, например $e_j = (Toyota, car)$. **Когипонимы**, такие как $(apple, pear)$, являются семантическими братьями – терминами с общим гиперонимом, например, “apple → fruit” и “pear → fruit”.

Отдельные узлы могут быть представлены как **синсет** – клика синонимов, таких как $\{car, vehicle, cars\}$ или $\{behemoth, hippopotamus\}$. Каждый узел представляет слово в заданном **смысле**. Удобнно задавать смысл указывая

гипероним смысла, например “jaguar (animal)” в отличие от “jaguar (car)”. Кроме этого, смысл может содержать текстовое **определение** его значения. В то же время, для **словоупотреблений** т.е. конкретных упоминаний слова “jaguar” в **текстовом корпусе** явным образом не заданы идентификаторы его смысла, такие как “animal” или “car”, за исключением **примеров** употребления слов в словарях. Вместе с графическими иллюстрациями смыслов, текстовые примеры, гиперонимы и определения помогают сделать смысл понятными для человека.

На Рис. 1-2 представлен обзор ключевых методов вычислительной лексической семантики, представленных в данной диссертации, и их взаимосвязей. На рисунке представлен текстовый контекст слова “jaguar”. Строка “jaguar” может относиться к нескольким значениям слова, например, “jaguar_(animal)” и “jaguar_(car)”, согласно описанию из составленной вручную **таксономии**. Однако в тексте отсутствует идентификатор, явно задающий смысл слова, такой как “car”. В следствие этого, возникает **неоднозначность** так как слово “jaguar” может относиться как к автомобилям (“car”), так и к животному (“animal”). Методы **разрешения неоднозначности смысла слова** более кратко именуемые методами **дизамбигуации** [5] автоматически определяют наиболее подходящее значение слова с учетом контекста. Другая постановка задачи разрешения неоднозначности смысла слова в контексте — **лексическая замена**. Вместо явного отождествления слов и идентификаторов их смыслов метод порождает семантически подходящие замены (обычно синонимы, когипонимы или гиперонимы) в правильном смысле. Например, в данном случае для слова “jaguar” подбираются замены на слова “puma”, “cheetah”, и “cougar” относящиеся к смыслу “animal”, а замены относящиеся к автомобильному смыслу исходного слова, такие как “Mercedes”, “BMW” или “Audi”, не предлагаются.

Цель выполнения дизамбигуации состоит в том, чтобы связать текст с точными и интерпретируемыми представлениями смысла слов из созданного

[5] Word Sense Disambiguation (WSD)

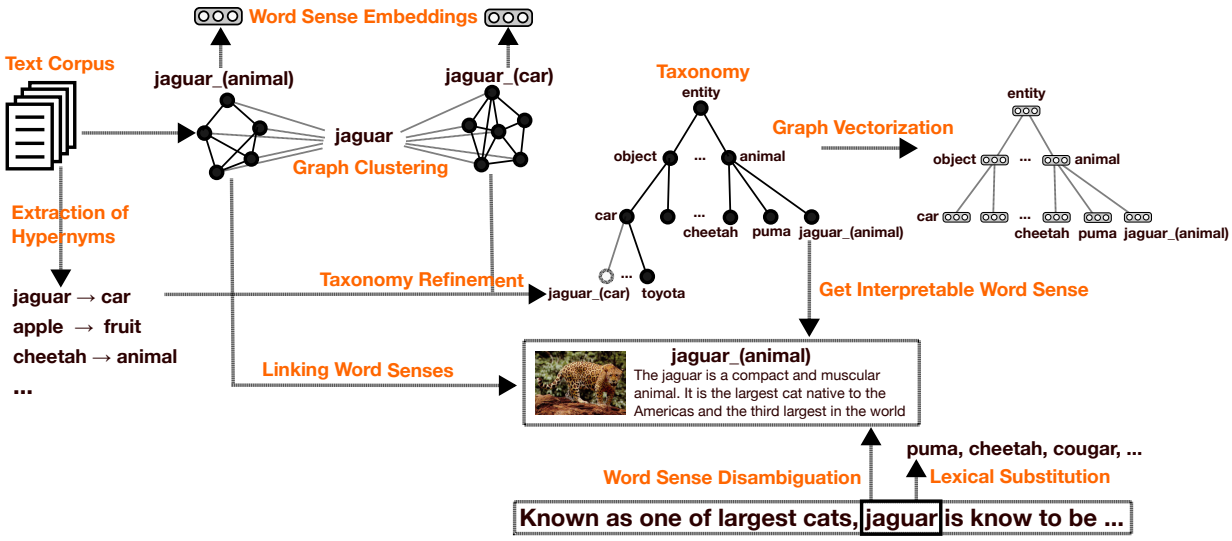

Рис. 1-2: Обзор представленных в диссертации методов вычислительной лексической семантики и их взаимосвязей.

вручную ресурса, который может содержать такие элементы, как определения, изображения, гиперонимы и связанные слова. В то же время, вновь извлеченные смыслы слов не имеют таких представлений. Вот почему одним из вкладов данной работы стало создание технологии автоматического построения интерпретируемых человеком представлений смыслов.

Использование представлений смыслов слов в моделях машинного обучения, особенно в нейронных сетях, затруднено, если они представлены в форме графов (которое соответствуют разреженному векторному представлению). Кроме этого, для вычислений сходства между смыслами слов низкоразмерный плотный векторный формат предпочтительнее разреженного векторного формата или графовых представлений. По этим причинам, в рамках данной работы были разработаны процедуры **векторизации графов** лексико-семантических ресурсов, таких как WordNet и DBpedia[6]. Для аналогичных целей были разработаны методы получения **векторов смысла слов**, которые используются для выполнения дизамбигуации на основе вычислений сходства между словами из контекста и этими векторными представлениями значений слов.

[6] `https://dbpedia.org`

Таким образом, на Рис. 1-2 представлен обзор различных методов вычислительной лексической семантики, предложенных в данной диссертации, и показано, как они взаимодействуют друг с другом. Материалы, представленные в диссертации, охватывают широкий круг задач, связанных с лексической вычислительной семантикой: они образуют прочную методологическую основу для обучения, пополнения, связывания, устранения неоднозначности и векторизации значений слов и отношений между ними.

С методологической точки зрения, большинство из разработанных методов и алгоритмов используют графовое представление (структуру данных). Алгоритмы кластеризации графов, в том числе вновь предложенные, используются для обработки лингвистических сетей различного типа. Использование представления графа вполне естественно, поскольку каждый лексико-семантический ресурс может быть представлен в виде графа, узлы которого представляют собой смыслы слов или термины, а ребра — семантические отношения между ними. Поскольку современные методы обработки текста в значительной степени полагаются на нейронные сети, работа с подобными лингвистическими графами потребовала их векторизации. С этой целью были разработаны методы эмбеддинга узлов лингвистических графов для решения различных задач, таких как пополнение лингвистических ресурсов и устранение смысловой неоднозначности слов.

Предложенные методы применимы ко всем распространенным лингвистическим ресурсам, таким как лексико-семантические базы данных, тезаурусы, таксономии, поскольку все они могут быть представлены в виде графа с узлами, соответствующими значениям слов. Важно отметить, что наиболее практическое применение сложного ресурса, такого как WordNet, предполагает сопоставление представлений смыслов, перечисленных в этом ресурсе, со словоупотреблениями в тексте. С этой целью, разработаны методы дизамбигуации поиска оптимальных соответствий с точки зрения семантической связности узлов лексического ресурса и слов в тексте.

Содержание диссертации разбито на 9 содержательных глав. В то же время,

каждая глава связана с использованием графовых и/или векторных представлений для задач лексико-семантической обработки. Среди всех методических приемов, кластеризацию графов следует выделить как центральную, применяемую в различных контекстах, таких как индукция значений слов, семантических фреймов или улучшение качества гиперонимических отношений. Таким образом, набор предлагаемых методов оперирует как с графовым так и векторным представлениями поскольку они в значительной степени дополняют друг друга.

Цели и задачи исследования

Целью диссертации является разработка методов вычислительной лексической семантики, которые позволили бы объединить достоинства (i) точных ручных интерпретируемых лексических ресурсов с низким лексическим охватом и (ii) неточных автоматический извлеченных неинтерпретируемых структур с высоким лексическим охватом. Для достижения этой цели, требуется произвести (i) разработку новых алгоритмов кластеризации больших лингвистических сетей, построенных как из вручную созданных лексических ресурсов, так и из графов, индуцированных из текста, (ii) разработку методов индукции лексико-семантических структур различного типа из текста, особенно значений слов и отношений гиперонимии, (iii) разработку методов, позволяющих интерпретировать индуцированные структуры так, как они находятся в созданных вручную ресурсах, (iv) разработку методов эффективного устранения неоднозначности в контексте относительно индуцированных смысловых представлений, (v) разработку эффективная векторизация лексико-семантических графов для использования в различных приложениях, (vi) разработку методов автоматического построения лексико-семантических иерархий.

Основные положения, выносимые на защиту:

1. Разработка алгоритма нечеткой кластеризации графов, эффективного и действенного для обработки больших лексико-семантических сетей

(Глава 2).

2. На основе разработанного алгоритма нечеткой кластеризации графов был предложен метод индукции трех лексико-семантических структур: синсетов, семантических фреймов и семантических классов (Глава 2).

3. Разработка метода построения векторных представлений смыслов слов из векторных представлений слов с использованием кластеризации графов и метода устранения неоднозначности смысла слов с использованием извлеченных представлений смыслов (Глава 3).

4. Разработка метода извлечения интерпретируемых смыслов слов (Глава 4).

5. Разработка структуры для связывания смыслов из лексических ресурсов и дистрибутивных моделей (Глава 5).

6. Разработка модели генерации гиперонимов слов на основе проекционного обучения с регуляризацией асимметрии отношений (Глава 6).

7. Предлагается метод постобработки отношений гиперонимии с использованием дистрибутивно индуцированных семантических классов: неправильные гиперонимы удаляются, а недостающие добавляются (Глава 7).

8. Разработка алгоритма построения таксономического дерева (состоящего из бинарных отношений гиперонимии между терминами) с использованием евклидовых и гиперболических (Пуанкаре) векторных представлений (Глава 8).

9. Разработка модели для построения векторных представлений узлов лингвистических сетей с использованием графовых метрик близости и приложения к задаче дизамбигуации (Глава 9).

10. Разработка методов нейронной лексической замены, которые объединяют информацию о целевом слове с информацией о контексте (Глава 10).

11. Исследование распределения лексико-семантических отношений позволило использовать нейронные модели лексического замещения (Глава 10).

Личный вклад автора включает формальную постановку задачи и постановку эксперимента для всех упомянутых выше результатов, разработку упомянутых методов и алгоритмов, анализ и обобщение результатов. Более конкретный вклад автора (и список ключевых соавторов), относящийся к каждой статье, представлен в списке публикаций ниже.

Новизна предлагаемого исследования заключается в разработке новых алгоритмов вычислительной лексической семантики. Новый подход к кластеризации графов лежит в основе нескольких недавно предложенных методов лексической семантики. Например, для автоматического создания лексико-семантических структур, таких как синсеты, семантические фреймы и семантические классы. Часть работ связаны с автоматической обработкой гиперонимических отношений между словами. В частности, в диссертации автор предлагает:

- Алгоритм нечеткой кластеризации графов (Глава 2);
- Методы создания синсетов, семантических классов и фреймов (Глава 2);
- Методы построения эмбеддингов смыслов слов (Глава 3);
- Метод построения интерпретируемых представлений слов (Глава 4);
- Метод связывания представлений смыслов извлеченных из текста со смыслами из лексических ресурсов (Глава 5);
- Модель для генерации гиперонимов (Глава 6);
- Метод постобработки отношений гиперонимии (Глава 7);
- Алгоритм построения таксономий (Глава 8);
- Метод векторизации узлов лингвистических графов (Глава 9);

- Методы нейронной лексической замены (Глава 10);
- Исследование распределения лексико-семантических отношений генерируемых нейронными моделями лексической замены (Глава 10).

Теперь обратимся к краткому изложению опубликованных исследований, на базе которых была подготовлена данная диссертация. Тема диссертации освещена в 42 публикациях [1–42], среди которых:

- 5 статей опубликовано на конференциях ранга CORE A* [3, 5, 9, 10, 13];
- 6 статей опубликовано на конференциях ранга CORE A [1, 2, 4, 7, 16];
- 5 статей опубликовано в журналах первого квартиля Q1 [6, 8, 11, 12, 15];
- 1 статья опубликована в студенческом треке конференции CORE A* [28];
- 1 статья опубликована в CORE A демо сессии конференции [18];
- 5 статей опубликовано на конференции CORE B [19, 20, 26, 27, 29];
- 11 статей опубликовано в сборниках основных томов конференций индексируемых Scopus [22–25, 30–34, 40, 41];
- 8 статей опубликовано на семинарах, проводимых совместно с ведущими конференциями (CORE A*/A) индексированных Scopus [17, 21, 35–39, 42].

Согласно Положению Диссертационного совета по компьютерным наукам ВШЭ и для краткости из 42 статей, соответствующих тематике диссертации, ниже перечислены только 20 ключевых, в том числе все 16 статей высшего уровня, т.е. A*/A/Q1 и 4 других уровней. Оставшиеся 22 публикации других уровней процитированы и находятся в открытом доступе. **Защита осуществляется на основании 14 публикаций из 20 перечисленных ниже работ.** А именно на основе первых 10 из списка публикаций высшего уровня [1–10] и 4 публикаций других уровней [17–20]. Для удобства ниже добавлены примечания если статья используется для защиты.

Публикации высшего уровня:

1. A. Panchenko, E. Ruppert, S. Faralli, S. P. Ponzetto, and C. Biemann, "**Unsupervised does not mean uninterpretable: The case for word sense induction and disambiguation**," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, (Valencia, Spain), pp. 86–98, Association for Computational Linguistics, Apr. 2017
   `https://aclanthology.org/E17-1009`
   [CORE A] *используется для защиты; главный соавтор;*

2. N. Arefyev, B. Sheludko, A. Podolskiy, and A. Panchenko, "**Always Keep your Target in Mind: Studying Semantics and Improving Performance of Neural Lexical Substitution**," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 1242–1255, International Committee on Computational Linguistics, Dec. 2020
   `https://aclanthology.org/2020.coling-main.107`
   [CORE A] *используется для защиты; главный соавтор;*

3. A. Kutuzov, M. Dorgham, O. Oliynyk, C. Biemann, and A. Panchenko, "**Making Fast Graph-based Algorithms with Graph Metric Embeddings**," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 3349–3355, Association for Computational Linguistics, July 2019
   `https://aclanthology.org/P19-1325`
   [CORE A*] *используется для защиты; главный соавтор;*

4. D. Ustalov, N. Arefyev, C. Biemann, and A. Panchenko, "**Negative Sampling Improves Hypernymy Extraction Based on Projection Learning**," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, (Valencia, Spain), pp. 543–550, Association for Computational Linguistics, Apr. 2017
   `https://aclanthology.org/E17-2087`
   [CORE A] *используется для защиты; главный соавтор;*

5. R. Aly, S. Acharya, A. Ossa, A. Köhn, C. Biemann, and A. Panchenko, "**Every Child Should Have Parents: A Taxonomy Refinement Algorithm Based on Hyperbolic Term Embeddings**," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 4811–4817, Association for Computational Linguistics, July 2019
   https://aclanthology.org/P19-1474
   [CORE A*] *используется для защиты; главный соавтор;*

6. D. Ustalov, A. Panchenko, C. Biemann, and S. P. Ponzetto, "**Watset: Local-Global Graph Clustering with Applications in Sense and Frame Induction**," *Computational Linguistics*, vol. 45, pp. 423–479, Sept. 2019
   https://aclanthology.org/J19-3002
   [Q1] *используется для защиты; главный соавтор;*

7. S. Faralli, A. Panchenko, C. Biemann, and S. P. Ponzetto, "**Linked Disambiguated Distributional Semantic Networks**," in *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II* (P. Groth, E. Simperl, A. J. G. Gray, M. Sabou, M. Krötzsch, F. Lécué, F. Flöck, and Y. Gil, eds.), vol. 9982 of *Lecture Notes in Computer Science*, pp. 56–64, 2016
   https://doi.org/10.1007/978-3-319-46547-0{_}7
   [CORE A] *используется для защиты; главный соавтор;*

8. C. Biemann, S. Faralli, A. Panchenko, and S. P. Ponzetto, "**A framework for enriching lexical semantic resources with distributional semantics**," *Nat. Lang. Eng.*, vol. 24, no. 2, pp. 265–312, 2018
   https://doi.org/10.1017/S135132491700047X
   [Q1] *используется для защиты; главный соавтор;*

9. D. Ustalov, A. Panchenko, and C. Biemann, "**Watset: Automatic Induction of Synsets from a Graph of Synonyms**," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Vancouver, Canada), pp. 1579–1590, Association for Computational Linguistics, July 2017
   https://aclanthology.org/P17-1145

[CORE A*] *используется для защиты; главный соавтор;*

10. D. Ustalov, A. Panchenko, A. Kutuzov, C. Biemann, and S. P. Ponzetto, "**Unsupervised Semantic Frame Induction using Triclustering**," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Melbourne, Australia), pp. 55–62, Association for Computational Linguistics, July 2018
    https://aclanthology.org/P18-2010
    [CORE A*] *используется для защиты; главный соавтор;*

11. Ö. Sevgili, A. Shelmanov, M. Y. Arkhipov, A. Panchenko, and C. Biemann, "**Neural entity linking: A survey of models based on deep learning**," *Semantic Web*, vol. 13, no. 3, pp. 527–570, 2022
    https://doi.org/10.3233/SW-222986
    [Q1] *главный соавтор;*

12. S. Anwar, A. Shelmanov, N. Arefyev, A. Panchenko, and C. Biemann, "**Text augmentation for semantic frame induction and parsing**," *Language Resources and Evaluation*, vol. 23, no. 3, pp. 527–556, 2023
    https://link.springer.com/article/10.1007/s10579-023-09679-8
    [Q1] *главный соавтор;*

13. A. Jana, D. Puzyrev, A. Panchenko, P. Goyal, C. Biemann, and A. Mukherjee, "**On the Compositionality Prediction of Noun Phrases using Poincaré Embeddings**," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 3263–3274, Association for Computational Linguistics, July 2019
    https://aclanthology.org/P19-1316
    [CORE A*] *главный соавтор;*

14. I. Nikishina, V. Logacheva, A. Panchenko, and N. Loukachevitch, "**Studying Taxonomy Enrichment on Diachronic WordNet Versions**," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 3095–3106, International Committee on Computational Linguistics, Dec. 2020

https://aclanthology.org/2020.coling-main.276

[CORE A] *главный соавтор;*

15. I. Nikishina, M. Tikhomirov, V. Logacheva, Y. Nazarov, A. Panchenko, and N. V. Loukachevitch, "**Taxonomy enrichment with text and graph vector representations**," *Semantic Web*, vol. 13, no. 3, pp. 441–475, 2022
https://doi.org/10.3233/SW-212955
[Q1] *главный соавтор;*

16. S. Faralli, A. Panchenko, C. Biemann, and S. P. Ponzetto, "**The ContrastMedium Algorithm: Taxonomy Induction From Noisy Knowledge Graphs With Just A Few Links**," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, (Valencia, Spain), pp. 590–600, Association for Computational Linguistics, Apr. 2017
https://aclanthology.org/E17-1056
[CORE A] *главный соавтор;*

Публикации других уровней:

17. M. Pelevina, N. Arefiev, C. Biemann, and A. Panchenko, "**Making Sense of Word Embeddings**," in *Proceedings of the 1st Workshop on Representation Learning for NLP*, (Berlin, Germany), pp. 174–183, Association for Computational Linguistics, Aug. 2016
https://aclanthology.org/W16-1620
[Scopus] *используется для защиты; главный соавтор;*

18. A. Panchenko, F. Marten, E. Ruppert, S. Faralli, D. Ustalov, S. P. Ponzetto, and C. Biemann, "**Unsupervised, Knowledge-Free, and Interpretable Word Sense Disambiguation**," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, (Copenhagen, Denmark), pp. 91–96, Association for Computational Linguistics, Sept. 2017
https://aclanthology.org/E17-1009
[CORE A, демо трек] *используется для защиты; главный соавтор;*

19. A. Panchenko, D. Ustalov, S. Faralli, S. P. Ponzetto, and C. Biemann, "**Improving Hypernymy Extraction with Distributional Semantic Classes**," in

Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018

`https://aclanthology.org/L18-1244`

[CORE B] *используется для защиты; главный соавтор;*

20. V. Logacheva, D. Teslenko, A. Shelmanov, S. Remus, D. Ustalov, A. Kutuzov, E. Artemova, C. Biemann, S. P. Ponzetto, and A. Panchenko, "**Word Sense Disambiguation for 158 Languages using Word Embeddings Only**," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, (Marseille, France), pp. 5943–5952, European Language Resources Association, May 2020

    `https://aclanthology.org/2020.lrec-1.728`

    [CORE B] *используется для защиты; главный соавтор;*

Доклады на конференциях и семинарах:

В этом разделе мы перечисляем конференции, на которых были представлены 42 статьи, относящиеся к теме этой диссертации, а не только для 14 статей, на которых основана защита.

1. **ACL-2019** [CORE A*] [3, 5, 13, 28, 36]: The 57th Annual Meeting of the Association for Computational Linguistics, (Флоренция, Италия), Association for Computational Linguistics, 2019.

2. **ACL-2018** [CORE A*] [10]: The 56th Annual Meeting of the Association for Computational Linguistics (Мельбурн, Австралия), Association for Computational Linguistics, 2018.

3. **ACL-2017** [CORE A*] [9]: The 55th Annual Meeting of the Association for Computational Linguistics (Ванкувер, Канада), Association for Computational Linguistics, 2017.

4. **ACL-2016** [CORE A*] [17]: The 54th Annual Meeting of the Association for Computational Linguistics (Берлин, Германия). Association for Computational Linguistics.

5. **IJCNLP-ACL-2021** [CORE A*] [35]: The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Бангкок, Тайланд), Association for Computational Linguistics.

6. **COLING-2022** [CORE A] [42]: The 29th International Conference on. Computational Linguistics (Кёнджу, Южная Корея), International Committee on Computational Linguistics, 2022.

7. **COLING-2020** [CORE A] [2, 14]: The 28th International Conference on Computational Linguistics, (Барселона, Испания), International Committee on Computational Linguistics, 2020.

8. **EACL-2017** [CORE A] [1, 4, 16, 39]: The 15th Conference of the European Chapter of the Association for Computational Linguistics (Валенсия, Испания), Association for Computational Linguistics, 2017. [1]

9. **EMNLP-2017** [CORE A] [18]: The 2017 Conference on Empirical Methods in Natural Language Processing (Копенгаген, Дания), Association for Computational Linguistics, 2017.

10. **ISWC-2016** [CORE A] [7]: The 15th International Semantic Web Conference, (Кобэ, Япония), vol. 9982 of Lecture Notes in Computer Science, 2016.

11. **NAACL-2019** [CORE A] [37, 38]: 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Минеаполис, США), Association for Computational Linguistics, 2019.

12. **NAACL-2016** [CORE A] [21]: The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Сан Диего, США), Association for Computational Linguistics.

13. **AACL-2022** [CORE B] [40]: The 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Таипей, Тайвань), Association for Computational Linguistics, 2022.

14. **LREC-2020** [CORE B] [20]: The 12th Language Resources and Evaluation Conference, (Марсель, Франция), European Language Resources Association, 2020.

15. **LREC-2018** [CORE B] [19, 26, 27]: The 11th International Conference on Language Resources and Evaluation (LREC 2018), (Миядзаки, Япония), European Language Resources Association (ELRA), 2018.

16. **LREC-2016** [CORE B] [29]: The 10th International Conference on Language Resources and Evaluation (LREC'16), (Порторож, Словения), pp. 2649–2655, European Language Resources Association (ELRA), 2016.

17. **PaM-2020** [Scopus] [22]: The Probability and Meaning Conference (Готенбург, Швеция), Association for Computational Linguistics, 2020.

18. **RANLP-2019** [Scopus] [33]: The International Conference on Recent Advances in Natural Language Processing (Варна, Болгария), INCOMA Ltd., 2019.

19. **GWC-2021** [Scopus] [41]: The 11th Global Wordnet Conference (Почефструм, ЮАР), Global Wordnet Association, 2021.

20. **AIST-2019** [Scopus/Q2] [32]: The 8th International Conference on Analysis of Images, Social Networks and Texts (Казань, Россия), July 2019, vol. 11832 of Lecture Notes in Computer Science, Springer.

21. **AIST-2017** [Scopus/Q2] [30]: The 6th International Conference on Analysis of Images, Social Networks and Texts (Москва, Россия), 2017, vol. 10716 of Lecture Notes in Computer Science, Springer.

22. **Dialogue-2018** [Scopus] [24, 25]: The 24th International Conference on Computational Linguistics and Intellectual Technologies (Москва, Россия), RGGU, June 2018.

23. **KONVENS-2018** [Scopus] [23]: The 14th Conference on Natural Language Processing (Вена, Австрия), September, 2018 Österreichische Akademie der Wissenschaften.

24. **KONVENS-2016** [Scopus] [31]: The 13th Conference on Natural Language Processing, (Бохум, Германия), 2016, Bochumer Linguistische Arbeitsberichte.

Глава 2

Кластеризация графов для индукции смыслов и фреймов

Материалы данной главы основаны на статьях [6, 9, 10] из списка 14 публикаций, на которых основана диссертация.

2.1 Введение

В этой главе мы рассмотрим задачу кластеризации графов, применяемую для извлечения различных лингвистических структур, таких как синсеты.

Пусть $G = (V, E)$ — неориентированный **граф**, где V — множество узлов, а $E \subseteq V^2$ — множество неориентированных ребер. Обозначим подмножество узлов $C^i \subseteq V$ как **кластер**. **Кластеризация графа** — это функция Cluster : $(V, E) \to C$ такая, что $V = \bigcup_{C^i \in C} C^i$. Существуют два класса кластеризации графов: алгоритмы **жесткой кластеризации** создают непересекающиеся кластеры, т. е. $C^i \cap C^j = \emptyset \iff i \neq j$, $\forall C^i, C^j \in C$, а **нечеткая кластеризация** допускает перекрытие кластеров, т. е. узел может быть членом нескольких кластеров в C.

Ниже представлен мета-алгоритм для нечеткой кластеризации графов. Он создает промежуточное представление входного графа, отражающее “неоднозначность” его узлов. Затем он использует жесткую кластеризацию для обнаружения кластеров в этом “дизамбигуированном” промежуточном графе.

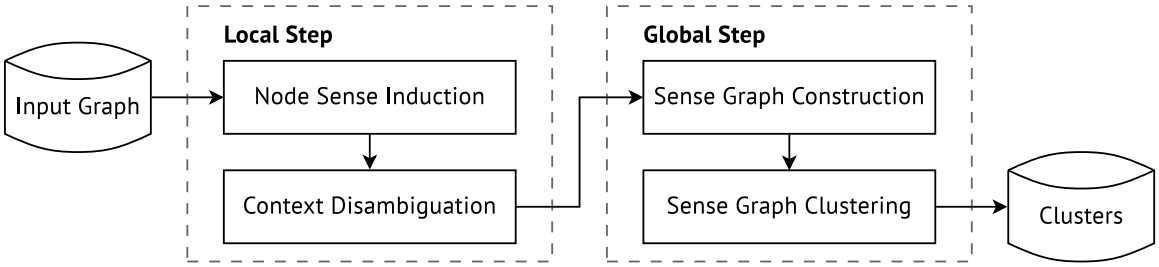

Рис. 2-1: Схема алгоритма, иллюстрирующая *локальный* этап индукции смысла узла и устранения неоднозначности контекста, а также *глобальный* этап построения и кластеризации дизамбигуированного (смыслового) графа.

2.2 Метод

В этом разделе представлен метаалгоритм кластеризации нечетких графов. Для заданного графа, соединяющий потенциально неоднозначные объекты, например слова, он создает набор однозначных перекрывающихся кластеров путем устранения неоднозначности и группировки неоднозначных объектов. Мета-алгоритм, который использует существующие алгоритмы *жесткой* кластеризации графов для получения *fuzzy* кластеризации, также известной как *нечеткая* кластеризация.

Алгоритм создает промежуточное представление входного графа, называемое *смысловым графом*. Это достигается путем индукции смысла узла, основанной на жесткой кластеризации окрестностей узлов входного графа. Смысловой граф имеет ребра, установленные между различными *смыслами* узлов входного графа. Глобальные кластеры входного графа получаются путем применения алгоритма жесткой кластеризации к смысловому графу.

Схема алгоритма изображена в Рис. 2-1: он принимает неориентированный граф $G = (V, E)$ в качестве входных данных и выводит набор кластеров C. Алгоритм состоит из двух шагов: локального и глобального. Локальный шаг устраняет неоднозначность в потенциально неоднозначных узлах в G. Глобальный шаг использует эти узлы с устранением неоднозначности для построения графа промежуточного смысла $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ и создания перекрывающейся кластеризации C. Watset параметризуется двумя алгоритмами разделения графа $\text{Cluster}_{\text{Local}}$ и $\text{Cluster}_{\text{Global}}$, а также мерой сходства контекста sim. Полный псевдокод Watset представлен в Алгоритме 1. Для иллюстрации при описании подхода будем приводить

Algorithm 1 Watset, локально-глобальный мета-алгоритм для нечеткой кластеризации графов.

Input: Граф $G = (V, E)$,
Жесткие алгоритмы кластеризации графов $\text{Cluster}_{\text{Local}}$ и $\text{Cluster}_{\text{Global}}$,
Мера контекстной близости $\text{sim} : (\text{ctx}(a), \text{ctx}(b)) \to \mathbb{R}$, $\forall\, \text{ctx}(a), \text{ctx}(b) \subseteq V$.
Output: Кластера C.

for all $u \in V$ **do** ▷ Локальный шаг: извлечение смыслов
  $\text{senses}(u) \leftarrow \emptyset$
  $V_u \leftarrow \{v \in V : \{u, v\} \in E\}$ ▷ Прим.: $u \notin V_u$
  $E_u \leftarrow \{\{v, w\} \in E : v, w \in V_u\}$
  $G_u \leftarrow (V_u, E_u)$
  $C_u \leftarrow \text{Cluster}_{\text{Local}}(G_u)$ ▷ Кластеризация смежных узлов с u
  for all $C_u^i \in C_u$ **do**
    $\text{ctx}(u^i) \leftarrow C_u^i$
    $\text{senses}(u) \leftarrow \text{senses}(u) \cup \{u^i\}$
$\mathcal{V} \leftarrow \bigcup_{u \in V} \text{senses}(u)$ ▷ Глобальный шаг: узлы смыслового графа
for all $\hat{u} \in \mathcal{V}$ **do** ▷ Локальный шаг: разрешение неоднозначностей
  $\widehat{\text{ctx}}(\hat{u}) \leftarrow \emptyset$
  for all $v \in \text{ctx}(\hat{u})$ **do**
    $\hat{v} \leftarrow \arg\max_{v' \in \text{senses}(v)} \text{sim}(\text{ctx}(\hat{u}) \cup \{u\}, \text{ctx}(v'))$ ▷ $\hat{u}$ является смыслом узла $u \in V$
    $\widehat{\text{ctx}}(\hat{u}) \leftarrow \widehat{\text{ctx}}(\hat{u}) \cup \{\hat{v}\}$
$\mathcal{E} \leftarrow \{\{\hat{u}, \hat{v}\} \in \mathcal{V}^2 : \hat{v} \in \widehat{\text{ctx}}(\hat{u})\}$ ▷ Глобальный шаг: дуги смыслового графа
$\mathcal{G} \leftarrow (\mathcal{V}, \mathcal{E})$ ▷ Глобальный шаг: построение смыслового графа
$\mathcal{C} \leftarrow \text{Cluster}_{\text{Global}}(\mathcal{G})$ ▷ Глобальный шаг: кластеризация смыслового графа
$C \leftarrow \{\{u \in V : \hat{u} \in \mathcal{C}^i\} \subseteq V : \mathcal{C}^i \in \mathcal{C}\}$ ▷ Удалить метки смыслов
return C

примеры со словами и их синонимами. Однако Watset не привязан только к лексическим единицам и отношениям, поэтому наши примеры приводятся *без потери общности*. Также обратите внимание, что Watset можно применять как для невзвешенных, так и для взвешенных графов, если базовые алгоритмы жесткой кластеризации $\text{Cluster}_{\text{Local}}$ и $\text{Cluster}_{\text{Global}}$ учитывают веса входного графа.

Локальный шаг алгоритма Watset извлекает смыслы узлов во входном графе и использует эту информацию, чтобы определить, какие именно смыслы узлов были соединены через ребра входного графа G.

Мы индуцируем смысл узлов, используя подход кластеризации окрестностей слов от [43]. В частности, мы предполагаем, что удаление узлов, участвующих во многих треугольниках, разделяет граф на несколько компонент связности. Каждый компонент соответствует смыслу целевого узла, поэтому эта процедура выполняется для каждого

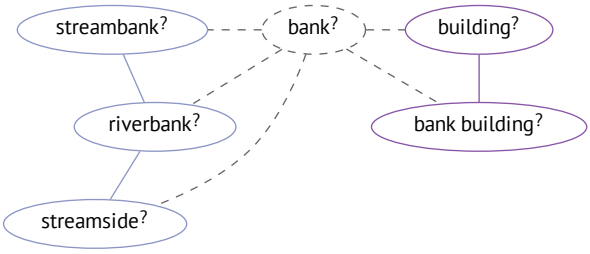

Рис. 2-2: Кластеризация окрестности узла “банк” входного графа приводит к получению двух кластеров, рассматриваемых как однозначные смысловые контексты: $bank^1 = \{streambank, riverbank, \dots\}$ и $\{bank^2 = bank\ building, building, \dots\}$.

Таблица 2.1: Пример извлеченных смыслов для узла “bank” и его соответствующие кластеры (контексты).

Смысл	**Контекст**
$bank^1$	$\{streambank, riverbank, \dots\}$
$bank^2$	$\{bank\ building, building, \dots\}$
$bank^3$	$\{bank\ company, \dots\}$
$bank^4$	$\{coin\ bank, penny\ bank, \dots\}$

узла независимо. Рис. 2-2 иллюстрирует этот подход к индукции смыслов.

Для каждого узла $u \in V$ мы извлекаем открытую окрестность $G_u = (V_u, E_u)$ из входного графа G, такую, что целевой узел u не входит в V_u (строки 3–5):

$$V_u = \{v \in V : \{u, v\} \in E\}, \tag{2.1}$$

$$E_u = \{\{v, w\} \in E : v, w \in V_u\}. \tag{2.2}$$

Затем мы запускаем алгоритм кластеризации жестких графов на G_u, который присваивает один узел одному и только одному кластеру, получая кластеризацию C_u (строка 6). Мы обрабатываем каждый полученный кластер $C_u^i \in C_u \subset V_u$ как представляющий контекст для различного смысла узла $u \in V$ (строки 7–9). Выполнение этой процедуры для всех слов в V приводит к набору смыслов для глобального шага (строка 10):

$$\mathcal{V} = \bigcup_{u \in V} \text{senses}(u). \tag{2.3}$$

Хотя на предыдущем шаге мы создали смыслы узлов и сопоставили их с соответствующими контекстами (Table 2.1), элементы этих контекстов не содержат смысловой информации. Например, контекст $bank^2$ в Рис. 2-3 состоит из двух

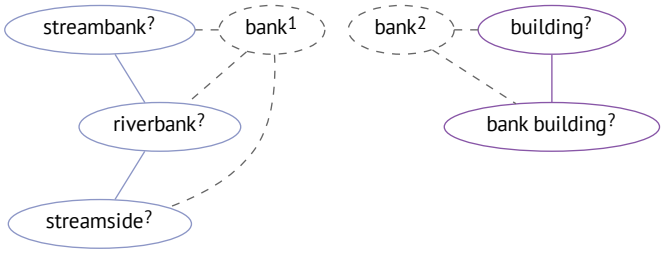

Рис. 2-3: Контексты для двух разных смыслов узла “банк”: только его смыслы: $bank^1$ и $bank^2$ в данный момент известны, в то время как неоднозначность остальных узлов в контексте еще предстоит разрешить.

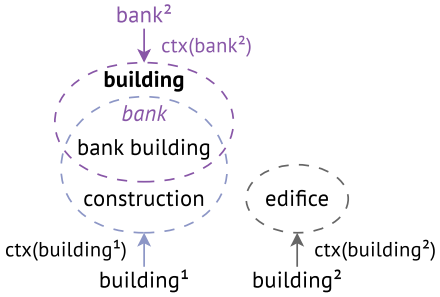

Рис. 2-4: Сопоставление значения неоднозначного узла “building” в контексте смысла $bank^2$. Для целей сопоставления слово “bank” временно добавляется в $\mathrm{ctx}(bank^2)$.

элементов: $\{bank\ Building^?, building^?\}$, смысловые метки которого на данный момент неизвестны. Мы восстанавливаем смысловые метки узлов в контексте, используя смысл, устраненный следующим образом.

Тогда, для смысла $\hat{u} \in \mathcal{V}$ узла $u \in V$ и контекста этого смысла $\mathrm{ctx}(\hat{u}) \subset V$, мы *устраняем неоднозначность* каждого узла $v \in \mathrm{ctx}(\hat{u})$. Для этого находим смысл $\hat{v} \in \mathrm{senses}(v)$, контекст $\mathrm{ctx}(\hat{v}) \subset V$ которого максимизирует сходство с целевым контекстом $\mathrm{ctx}(\hat{u})$. Мы вычисляем сходство, используя меру сходства контекста $\mathrm{sim} : (\mathrm{ctx}(a), \mathrm{ctx}(b)) \to \mathbb{R}$, $\forall\, \mathrm{ctx}(a), \mathrm{ctx}(b) \subseteq V$. Типичным выбором меры сходства являются скалярное произведение, косинусное сходство, индекс Жаккара и т. д. Следовательно, мы *устраняем неоднозначность* каждого элемента контекста $v \in \mathrm{ctx}(\hat{u})$:

$$\hat{v} = \arg\max_{v' \in \mathrm{senses}(v)} \mathrm{sim}(\mathrm{ctx}(\hat{u}) \cup \{u\}, \mathrm{ctx}(v')). \tag{2.4}$$

Пример на рисунке 2-4 иллюстрирует процесс устранения неоднозначности смысла узла.

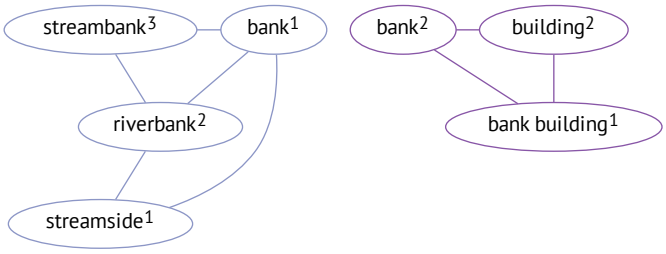

Рис. 2-5: Кластеризация *смыслового графа* $\mathcal{G}$ дала два кластера, $\{bank^1, streambank^3, riverbank^2, \dots\}$ и $\{bank^2, bank\ building^1, building^2, \dots\}$; если убрать метки смыслов кластеры будут пересекатся, в результате чего получится нечеткая кластеризация входного графа G.

В итоге мы строим дизамбигуированный контекст $\widehat{\text{ctx}}(\hat{u}) \subset \mathcal{V}$ который является версией $\text{ctx}(\hat{u})$ с идентификаторами смысла. Мы используем процедуру дизамбигуации заданную формулой (2.4) для каждого узла $v \in \text{ctx}(\hat{u})$:

$$\widehat{\text{ctx}}(\hat{u}) = \{\hat{v} \in \mathcal{V} : v \in \text{ctx}(\hat{u})\}. \tag{2.5}$$

Глобальный шаг алгоритма создает промежуточный *смысловой граф*, выражающий связи между узлами, обнаруженными на локальнмо шаге. Мы предполагаем, что узлы $\mathcal{V}$ смыслового графа однозначны, поэтому запуск алгоритма жесткой кластеризации на этом графе дает на выходе кластеры C, покрывающие множество узлов V входного графа G .

Используя набор смыслов узлов, определенный в формулой (2.3), мы строим граф смыслов $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, устанавливая ненаправленные ребра между смыслы, связанные через неоднозначные контексты (lines 16–17):

$$\mathcal{E} = \{\{\hat{u}, \hat{v}\} \in \mathcal{V}^2 : \hat{v} \in \widehat{\text{ctx}}(\hat{u})\}. \tag{2.6}$$

Запуск алгоритма жесткой кластеризации в $\mathcal{G}$ создает набор сенсорных кластеров $\mathcal{C}$, каждый из которых $\mathcal{C}^i \in \mathcal{C}$ является подмножеством $\mathcal{V}$ (line 18). Чтобы получить набор кластеров C, покрывающий множество узлов V входного графа G, мы просто удалим смысловые метки с элементов кластеров $\mathcal{C}$ (line 19):

$$C = \big\{\{u \in V : \hat{u} \in \mathcal{C}^i\} \subseteq V : \mathcal{C}^i \in \mathcal{C}\big\}. \tag{2.7}$$

Таблица 2.2: Различные типы входных лингвистических графов, кластеризованные алгоритмом Watset и соответствующие индуцированные выходные символические лингвистические структуры.

Входные узлы графа	**Входные дуги графа**	**Выходная лингвистическая структура**
многозначные слова	отношения синонимии	синсеты
тройки (подлежащее, глагол, сказуемое)	наиболее схожие по смыслу тройки	семантические фреймы
многозначные слова	наиболее схожие по смыслу слова	семантические классы

Рис. 2-5 иллюстрирует смысловой граф и его кластеризацию на примере узла “bank”. Построение смыслового графа требует устранения неоднозначности узлов входного графа. Обратите внимание, что традиционные подходы к смысловой индукции на основе графов, такие как предложенные [44–46], не выполняют этот шаг, а выполняют только локальную кластеризацию графа, поскольку они не нацелены на глобальное представительство кластеров.

В результате глобального шага с помощью промежуточного смыслового графа $\mathcal{G}$ получается набор кластеров C входного графа G. Представленный подход к локально-глобальной кластеризации графов, Watset, позволяет естественным образом добиться *нечеткой* кластеризации графа, используя только *жесткие* алгоритмы кластеризации.

Стоит отметить, что оригинальная статья [6] также включает описание упрощенной версии Watset, которая позволяет построить смысловой граф $\mathcal{G}$ за линейное время $O(|E|)$ путем запроса индекса смысла узла для устранения неоднозначности входных ребер E детерминированным способом. Остальные шаги идентичны оригинальному алгоритму Watset, представленному здесь.

2.3 Результаты

Эксперименты показывают, что алгоритм показывает высокие результаты в трех приложениях: индукция синсетов из графов синонимии, индукция семантических фреймов из графов синтаксических зависимостей и индукция семантических классов из дистрибутивного тезауруса. Алгоритм является универсальным и может

Размер	**Синсет**
2	decimal point, dot
2	wall socket, power point
3	gullet, throat, food pipe
3	CAT, computed axial tomography, CT
4	microwave meal, ready meal, TV dinner, frozen dinner
4	mock strawberry, false strawberry, gurbir, Indian strawberry
5	objective case, accusative case, oblique case, object case, accusative
5	discipline, sphere, area, domain, sector
6	radio theater, dramatized audiobook, audio theater, radio play, radio drama, audio play
6	integrator, reconciler, consolidator, mediator, harmonizer, uniter
7	invite, motivate, entreat, ask for, incentify, ask out, encourage
7	curtail, craw, yield, riding crop, harvest, crop, hunting crop

Таблица 2.3: Примеры извлечения синсинов с помощью предложенного метода кластеризации из графа неоднозначных синонимов.

применяться и к другим графам лингвистических данных (см. Таблицу 2.2). Примеры синсетов, созданных методом Watset[MCL, MCL], представлены в Таблице 2.3.

Более подробную информацию, включая теоретический и экспериментальный анализ сложности алгоритмов и приложения к реальным графам, можно найти в [6, 9, 10].

Глава 3

Векторные представления смыслов

Материалы данной главы основаны на статьях [17, 20] из списка 14 публикаций на которых основана диссертация.

3.1 Введение

В этой главе кластеризация графов применяется к лексико-семантическим сетям для получения векторных представлений (эмбеддингов) значений слов.

Задача обучения эмбеддингов смысла слова, рассматриваемая в этой главе, заключается в следующем. Входные данные представляют собой набор **векторов слов** (эмбеддингов слов) из словаря V: $\forall v \in V \ \exists \mathbf{v} \in \mathbb{R}^d$, где d — размерность векторного пространства. Выходными данными задачи являются (i) **инвентарь смыслов слов** S: $\forall v \in V \ \exists \{s_1, ..., s_k\} : s_i \subset V$, где k — количество значений слова v и (ii) набор **векторов значений слов** $\forall s_i \ \exists \mathbf{s}_i \in \mathbb{R}^d$.

Представляется простой, но эффективный подход для построения подобных эмбеддингов смыслов слов. В отличие от существующих методов, которые либо строят смысловые эмбеддинги из корпусов, либо полагаются на заданный в лексическом ресурсе инвентарь смыслов, предложенный подход может постороить инвентарь смыслов с использованием предобученных эмбеддингов слов посредством кластеризации эго-сетей связанных слов. Предложенный механизм разрешения неоднозначности слова в контексте (дизамбигуации) позволяет размечать упоминания слов в текстах с использованием индуцированного инвентаря смыслов.

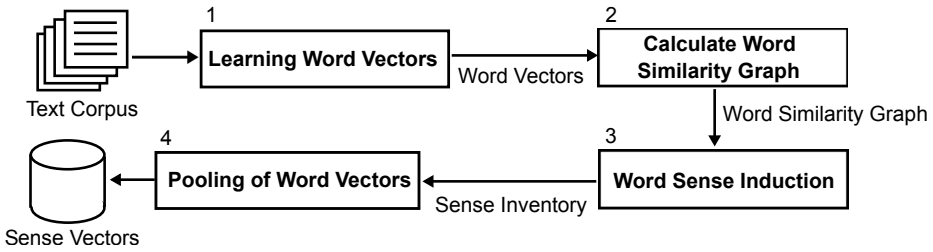

Рис. 3-1: Обзор метода построения эмбеддингов смыслов слов SenseGram.

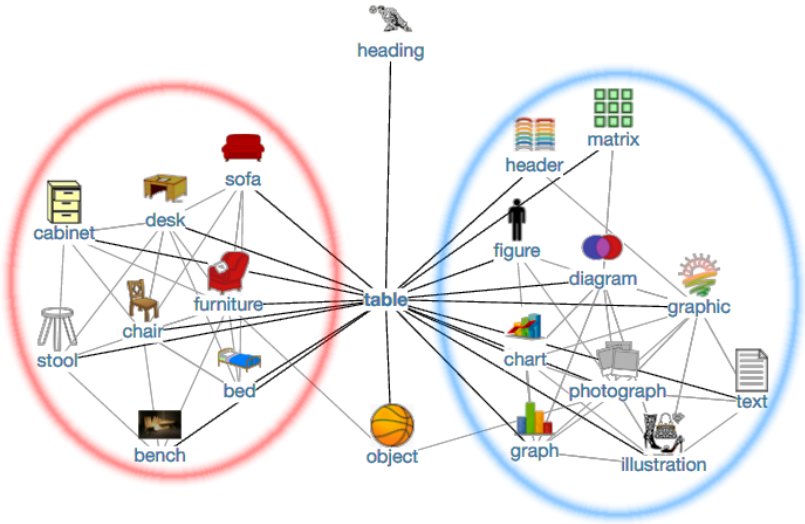

Рис. 3-2: Визуализация эго-сети слова “table"с кластерами соответствующими смыслам “furniture” и “data”. При этом, целевое слово (эго) “table"исключается из эго-сети при кластеризации.

3.2 Метод

Метод состоит из четырех основных этапов, изображенных на Рис. 3-1: (1) построение эмбеддингов слов; (2) построение графа ближайших соседей на основе подобия векторов слов; (3) индукция значений слов с использованием кластеризации эго-сетей; и (4) агрегирование векторов слов по индуцированным значениям. Метод может использовать предобученные эмбеддинги слов – в этом случае этап (1) не требуется.

Обучение векторов слов

Для получение векторов слов используется *word2vec* [47] и fastText [48], но возможно использовать аналогичные предобученные векторные представления слов. Окончательные эмбеддинги смыслов остаются в том же векторном пространстве, что

и эти входные векторы слов.

Расчёт графа ближайших соседей слов

На этом этапе производится построение графа семантических ближайших соседей для слов, состоящих из таких ребер как (table, Desk, 0.78). Для каждого слова мы извлекаем 200 его ближайших соседей. Этот граф рассчитывается либо на основе эмбеддингов слов, полученных на предыдущем шаге, либо с использованием значений семантической близости полученных с помощью дистрибутивной модели *JoBimText* [49].

Algorithm 2 Индукция смыслов слов: базовый алгоритм.

Вход : T – граф ближайших соседей слов, N – размер эго-сети, n – степень связности эго-сети, k – минимальный размер кластера

Выход: для каждого слова $t \in T$, кластеризация S_t его N ближайших соседей

```
foreach t ∈ T do
    V ← N most similar terms of t from T
    G ← graph with V as nodes and no edges E
    foreach v ∈ V do
        V' ← n most similar terms of v from T
        foreach v' ∈ V' do
            if v' ∈ V then add edge (v, v') to E
        end
    end
    S_t ← ChineseWhispers(G)
    S_t ← {s ∈ S_t : |s| ≥ k}
end
```

Индукция смыслов слов: базовый алгоритм

Способ извлечения смыслов слов описан в Алгоритме 2. За одну итерацию обрабатывается одно слово t из графа подобия слов T. Сначала извлекаются узлы V эго-сети G: это N наиболее похожие слова t по T. Целевое слово (эго) t само по себе не является частью эго-сети. Затем соединяются узлы в G с их n наиболее похожими словами из T. В заключение, эго-сеть кластеризуется с помощью алгоритма Chinese Whispers [50]. Этот метод не имеет параметров, поэтому мы не делаем предположений о количестве значений для заданного слова.

Алгоритм индукции смыслов имеет три мета-параметра: размер эго-сети целевого эго-слова t (N) ; связность эго-сети (n) – это максимальное количество ребер, которое разрешено иметь соседу эго v внутри эго-сети; минимальный размер кластера k.

Параметр n регулирует степень детализации инвентаря смыслов. В экспериментах были установлены значения $N = 200$, $n = \{50, 100, 200\}$ и $k = \{5, 15\}$, чтобы получить разное среднее количество смыслов в инвентаре. Каждое слово в смысловом кластере имеет вес, равный семантической близости между этим словом и неоднозначным словом t.

Индукция смыслов слов: улучшенный алгоритм

Улучшенная процедуру построения графа использует операции сложения и вычитания векторов в пространстве эмбеддингов слов [51], тогда как базовый алгоритм полагается только на список ближайших соседей в пространстве эмбеддингов слов. Построение графа ближайших соседей происходит с помощью способа аналогичного базовому. Однако, в дополнение, производится фильтрация узлов графа с помощью описанной ниже процедуры:

1. Построение списка $\mathcal{N} = \{w_1, w_2, ..., w_N\}$ из N ближайших соседей для целевого слова w (эго).

2. Построение списка $\Delta = \{\delta_1, \delta_2, ..., \delta_N\}$ для каждого w_i в $\mathcal{N}$, где $\delta_i = w - w_i$. Векторы в δ содержат компоненты смысла w, не связанные с соответствующими ближайшими соседями из $\mathcal{N}$.

3. Построение списка $\overline{\mathcal{N}} = \{\overline{w_1}, \overline{w_2}, ..., \overline{w_N}\}$, такого что $\overline{w_i}$ находится среди ближайших соседей δ_i в векторном пространстве. Другими словами, $\overline{w_i}$ — это слово, которое наиболее похоже на целевое слово w и наименее похоже на своего соседа w_i. Мы называем $\overline{w_i}$ **анти-узлом** w_i. Набор N ближайших соседей и их анти-узлов образуют набор *анти-ребер*, т.е. пары наиболее разнородных узлов — тех, которые не должны быть соединены в графе ребром: $\overline{E} = \{(w_1, \overline{w_1}), (w_2, \overline{w_2}), ..., (w_N, \overline{w_N})\}$.

   Чтобы прояснить это, рассмотрим целевое слово $w = python$, наиболее подобное ему слово $w_1 = Java$ и анти-узел $\overline{w_i} = Snake$, который является наиболее близким соседом для $\delta_1 = w - w_1$. Вместе они образуют **анти-ребро** $(w_i, \overline{w_i}) = (Java, snake)$, состоящее из пары семантически далеких слов.

4. Построение V – множества вершин семантического графа $G = (V, E)$ из списка анти-ребер $\overline{E}$, с помощью следующей рекуррентной процедуры:

$V = V\ cup\{w_i, \overline{w_i} : w_i \in \mathcal{N}, \overline{w_i} \in \mathcal{N}\}$, т.е. мы добавляем слово из списка ближайших соседей u его анти-узел только в том случае, если они оба являются ближайшими соседями исходного слова w. Мы не добавляем ближайших соседей w, если их анти-узлы не принадлежат $\mathcal{N}$. Таким образом, мы добавляем только слова, которые могут помочь дискриминировать разные значения w.

5. Построение множества ребер E. Для каждого $w_i \in \mathcal{N}$ извлекается K ближайших соседей $\mathcal{N}'_i = \{u_1, u_2, ..., u_K\}$ и определяется $E = \{(w_i, u_j) : w_i \in V, u_j \in V, u_j \in \mathcal{N}'_i, u_j \neq \overline{w_i}\}$. Таким образом, мы удаляем ребра между словом w_i и его ближайшим соседом u_j, если u_j также является его анти-узлом. Предполагается, что w_i и $\overline{w_i}$ принадлежат разным смыслам w, поэтому они не должны быть связаны (т.е. мы никогда не добавляем анти-ребра в E). Поэтому любую связь между ними мы считаем ошибочной и удаляем ее.

После построения графа выполняется кластеризация с использованием алгоритма Chinese Whispers [45]. На Рис. 3-3 показан пример графа для слова *Ruby* для $N = 50$ ближайших соседей построенного эмбеддингов fastText.

Вычисление векторов смыслов

На этом этапе производятся вычисления эмбеддингов для каждого смысла из индуцированного инвентаря. В данном методе делается предположение о том, что смысл слова – это совокупность слов, передающих этот смысл. Вектор смысла определяется как функция векторов слов, представляющих элементы кластера. Пусть W — набор всех слов в обучающем корпусе, а $S_i = \{w_1, \ldots, w_n\} \subseteq W$ — кластер смысла, полученный на предыдущем шаге. Рассмотрим функцию $vec_w : W \to \mathbb{R}^m$, которая отображает слова в их векторы, и функцию $\gamma_i : W \to \mathbb{R}$, которая отображает кластер слов в их вес. в кластере S_i. Были протестированы два способа расчета векторов смыслов: невзвешенное среднее векторов слов:

$$\mathbf{s}_i = \frac{\sum_{k=1}^{n} vec_w(w_k)}{n}; \tag{3.1}$$

и взвешенное среднее векторов слов:

$$\mathbf{s}_i = \frac{\sum_{k=1}^{n} \gamma_i(w_k) vec_w(w_k)}{\sum_{k=1}^{n} \gamma_i(w_k)}. \tag{3.2}$$

Вектор	Ближайшие соседи
table	tray, bottom, diagram, bucket, brackets, stack, basket, list, parenthesis, cup, trays, pile, playfield, bracket, pot, drop-down, cue, plate
table#0	leftmost#0, column#1, randomly#0, tableau#1, top-left#0, indent#1, bracket#3, pointer#0, footer#1, cursor#1, diagram#0, grid#0
table#1	pile#1, stool#1, tray#0, basket#0, bowl#1, bucket#0, box#0, cage#0, saucer#3, mirror#1, birdcage#0, hole#0, pan#1, lid#0

Таблица 3.1: Ближайшие соседи вектора слова “table"и векторов его смыслов, созданных с помощью нашего метода.

Таблица 3.1 представляет пример результатов взвешенного усреднения векторов.

Разрешение неоднозначности смысла слов

В этом разделе описывается, как эмбеддинги смыслов используются для разрешения неоднозначности значения слова в контексте. Для целевого слова w и его контекста $C = \{c_1, \ldots, c_k\}$ производится загрузка его эмбеддингов смыслов в соответствии с инвентарем: $S = \{\ mathbf{s}_1, \ldots, \mathbf{s}_n\}$. Мы используем две стратегии для выбора правильного смысла. Первый основан на подсчете вероятности смысла в данном контексте:

$$s^* = \arg\max_i P(C|\mathbf{s}_i) = \arg\max_i \frac{1}{1 + e^{-\bar{\mathbf{c}}_c \cdot \mathbf{s}_i}}, \tag{3.3}$$

где $\bar{\mathbf{c}}_c$ — среднее значение эмбеддингов контекста: $k^{-1} \sum_{i=1}^{k} vec_c(c_i)$ и функция $vec_c : W \to \mathbb{R}^m$ отображают слова контекста в эмбеддинги контекста.

Вторая стратегия устранения неоднозначности основана на сходстве смысла и контекста:

$$s^* = \arg\max_i sim(\mathbf{s}_i, C) = \arg\max_i \frac{\bar{\mathbf{c}}_w \cdot \mathbf{s}_i}{\|\bar{\mathbf{c}}_w\| \cdot \|\mathbf{s}_i\|}, \tag{3.4}$$

где $\bar{\mathbf{c}}_w$ — среднее значение эмбеддингов слов: $\bar{\mathbf{c}}_w = k^{-1} \sum_{i=1}^{k} vec_w(c_i)$. Последний метод использует только векторы слов (vec_w) и не требует векторов контекста (vec_c).

Обычно только несколько слов в контексте имеют значение для устранения смысловой неоднозначности, например, слова“chairs"и “kitchen"обозначают слово “table” в “They bought a table and chairs for kitchen". Для каждого слова c_j в контексте $C = \{c_1, \ldots, c_k\}$ мы вычисляем оценку, которая количественно определяет, насколько

хорошо оно дискриминирует смыслы:

$$\max_i f(\mathbf{s}_i, c_j) - \min_i f(\mathbf{s}_i, c_j), \tag{3.5}$$

где $\mathbf{s}_i$ – смысл неоднозначного слова, а f — одна из стратегий устранения неоднозначности: либо $P(c_j|\mathbf{s}_i)$, либо $sim(\ mathbfs_i, c_j)$. Для устранения неоднозначности используются p наиболее дискриминативных контекстных слов.

Разметка индуцированных смыслов

Мы размечаем каждую группу слов, обозначающую смысл, чтобы сделать их и результаты дизамбигуации более понятными для людей. В главе ниже мы покажем, как можно использовать гиперонимы для разметки кластеров, такие как "animal" в "python (animal)". Однако для некоторых языков с ограниченными ресурсами гиперонимы недоступны. Поэтому мы описываем более простой метод выбора ключевого слова, которое помогает интерпретировать каждый кластер. Для каждого узла графа $v \in V$ подсчитаем количество анти-ребер, к которым он принадлежит:

$$keyness(v) = |\{(w_i, \overline{w_i}) : (w_i, \overline{w_i}) \in \overline{E} \wedge (v = w_i \vee v = \overline{w_i})\}|. \tag{3.6}$$

Кластеризация графа дает разбиение V на n кластеры: $V = \{V_1, V_2, ..., V_n\}$. Для каждого кластера V_i определим *ключевое слово* w_i^{key} как слово с наибольшим числом анти-ребер $keyness(\cdot)$ среди слов в этом кластере.

3.3 Результаты

Основные эксперименты проводились для английского языка. Кроме того, этот метод использовался для создания коллекции смысловых инвентарей для 158 языков на основе исходных предварительно обученных векторов слов fastText [52], что позволило реализовать метод дизамбигуации на этих языках.

Результаты экспериментов на наборах данных извлечения смыслов и задаче лексического сходства, что производительность метода сопоставима с современными методами дизамбигуации без учителя. Детали экспериментальных результатов и их анализ можно найти в [17, 20].

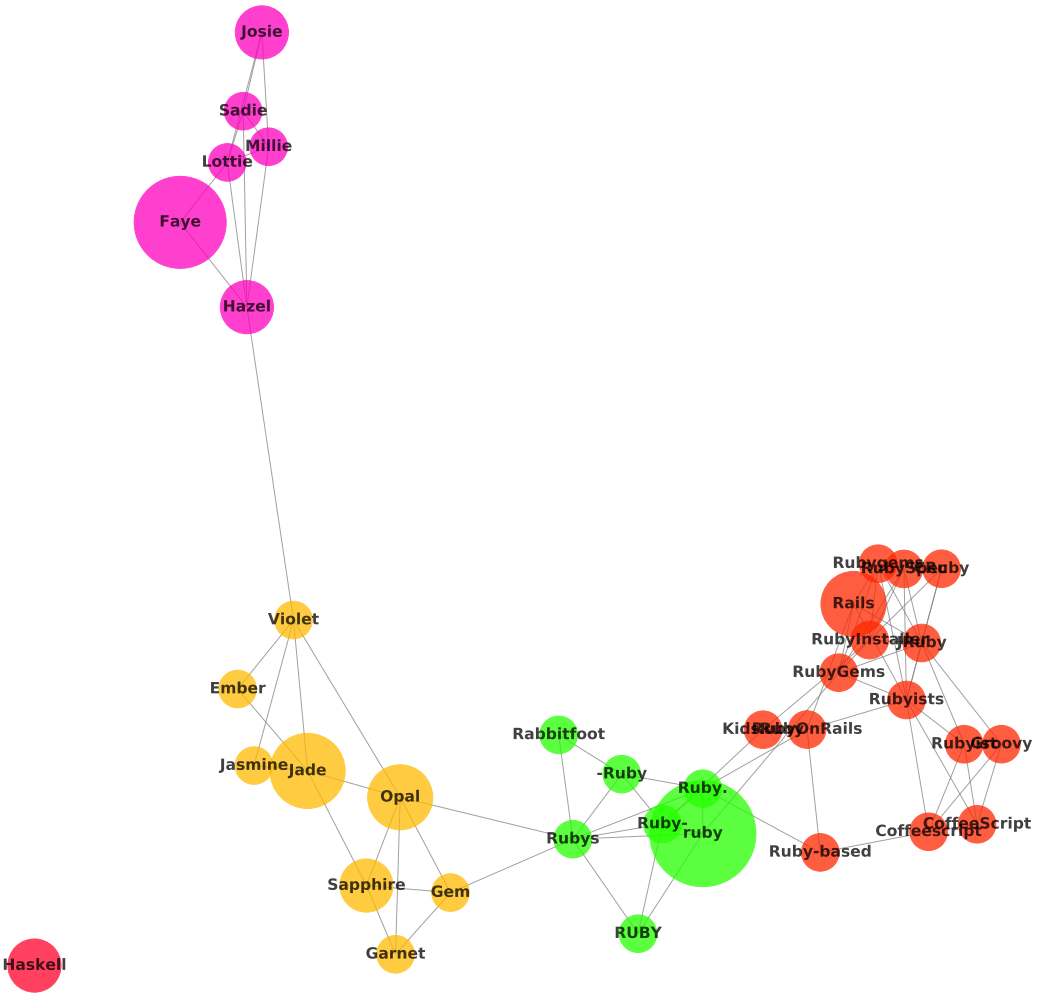

Рис. 3-3: Граф ближайших соседей слова *Ruby* разделен по нескольким значениям: языки программирования, женские имена, драгоценные камни, различные варианты написания слова *Ruby*. Размер узла обозначает важность слова, при этом самый большой узел в кластере используется в качестве ключевого слова для интерпретации индуцированного смысла слова.

Глава 4

Интерпретируемые представления смыслов и дизамбигуация

Материалы данной главы основаны на статьях [1] и [18] из списка 14 публикаций на которых основана диссертация.

4.1 Введение

В данной главе предлагается метод для индукции интерпретируемых смыслов слов и их дизамбигуации, основанный на методах, представленных в двух предыдущих главах.

Входными данными для задачи, рассмотренной в данной главе, являются (i) **инвентарь смыслов слов** S и (ii) **упоминание** слова v в **контексте** C. Выходными данными являются (i) **идентификатор смысла** слова v, соответствующий упоминанию в контексте C, и (ii) **интерпретируемое представление** смысла слова s.

Представленный метод является интерпретируемым на трех уровнях: инвентаря смыслов слов, контекстных признаков смыслов и процедуры дизамбигуации.

Интерпретируемость статистической модели важна, поскольку она позволяет нам понять причины ее прогнозов [53–55]. Интерпретируемость моделей дизамбигуации (1) позволяет пользователю понять, почему в данном контексте наблюдается тот или иной смысл (например, для образовательных приложений); (2) выполняет всесторонний анализ правильных и ошибочных прогнозов, что приводит к улучшению

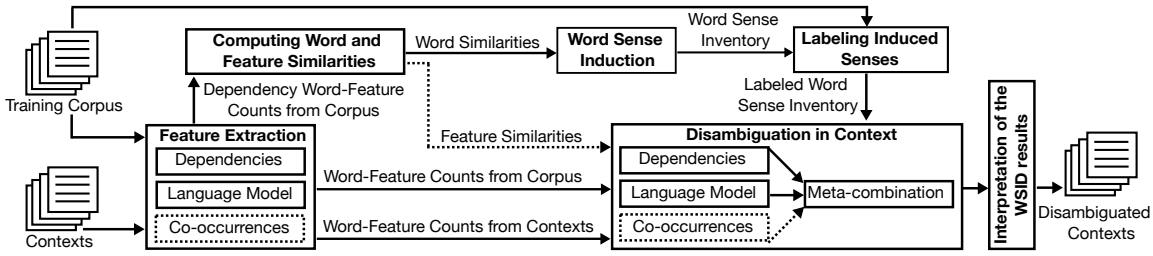

Рис. 4-1: Схема представленного метода для извлечения смыслов и дизамбигуации.

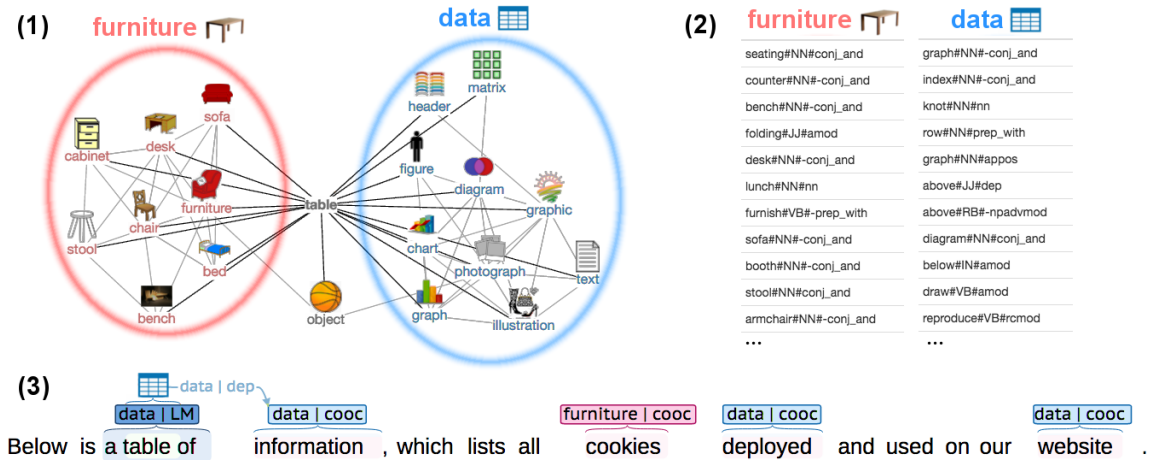

Рис. 4-2: Интерпретация значений слова “table” на трёх уровнях: (1) инвентарь смыслов слова; (2) контекстные признаки; (3) метод дизамбигуации. Смысловые метки (“furniture” и “data”) получаются автоматически на основе разметки кластеров гипернимами. Изображения, связанные со смыслами, извлекаются с помощью запросов в поисковую систему: “table data” и “table furniture”.

моделей устранения неоднозначности.

Целью этой главы является интерпретируемый метод дизамбигуации. Новизна нашего метода заключается в (1) технике устранения неоднозначности, которая опирается на индуцированные инвентаризации в качестве основы для изучения представлений смысловых признаков, (2) технике, позволяющей интерпретировать индуцированные смысловые репрезентации путем разметки их гиперонимами и изображениями.

4.2 Метод

Метод состоит из пяти этапов, проиллюстрированных на рисунке 4-1: извлечение признаков контекста; вычисление сходства слов и признаков; индукция смыслов слов; разметка кластеров гиперонимами и изображениями; дизамбигуация слов в контексте на основе индуцированного инвентаря, и интерпретация результатов метода дизамбигуации.

Извлечение контекстных признаков слов Цель этого шага — извлечь матрицу слово-признак из корпуса текстов. В частности, мы извлекаем функции на основе синтаксических зависимостей, совместной встречаемости слов и из статистических языковых моделей. Также рассчитывается графы подобия слов и признаков.

Извлечение смыслов слов Инвентарь смыслов слов извлекается методом кластеризации эко-сетей графов дистрибутивно подобных слов. Инвентарь представляет смыслы с помощью кластеров слов, таких как, “chair, bed, bench, stool, sofa, desk, cabinet"для смысла “furniture” слова “table”.. Метод индукции смыслов обрабатывает одно слово t графа подобия слов T за одну итерацию. Сначала производится извлечение узлов эго-сети G слова t, представляющие собой N наиболее похожих слов t согласно T (см. Рис. 4-2 (1)). При этом, целевое слово t само по себе не является частью эго-сети. Каждый узел в G соединяется n наиболее близкими словами согласно T. В заключение, эго-сеть кластеризуется алгоритмом [50]. Параметр n регулирует степень гранулярности инвентаря смыслов: были протестированы значения $n \in \{200, 100, 50\}$ и $N = 200$.

Разметка извлеченных смыслов гиперонимами и картинками

Для улучшения интерпретируемости извлеченных смыслов слов, каждому слову в кластере сопоставляется изображение (см. Рис. 4-2). Отправляется запрос в систему поиска изображений, используя запрос, состоящий из целевого слова и его гиперонима, например “jaguar car”. Наиболее релевантное изображение выбирается для представления индуцированного смысла слова.

Дизамбигуация с использованием индуцированного инвентаря смыслов

Для того чтобы устранить неоднозначность целевого слова t в контексте, извлекаются контекстные признаки C и передаются в алгоритм 3. Из всех смыслов

Algorithm 3 Метод дизамбигуации на базе индуцированного инвентаря смыслов слов.

input : Слово t, контекстные признаки C, инвентарь смыслов I, матрица слово-признак F, использование отката к наибольшему кластеру LCB, использование расширения признаков FE.

output: Sense of the target word t in inventory I and confidence score.

$S \leftarrow \mathsf{getSenses}\ (I, t)$
if FE **then**
  $C \leftarrow \mathsf{featureExpansion}(C)$
end
foreach $(sense, cluster) \in S$ **do**
  $\alpha[sense] \leftarrow \{\}$
  foreach $w \in cluster$ **do**
    foreach $c \in C$ **do**
      $\alpha[sense] \leftarrow \alpha[sense] \cup F(w, c)$
    end
  end
end
if $\max_{sense \in S}\ mean(\alpha[sense]) = 0$ **then**
  if LCB **then**
    return $\arg\max_{(\_, cluster) \in S} |cluster|$
  else
    return -1 `// reject to classify`
  end
else
  return $\arg\max_{(sense, \_) \in S} mean(\alpha[sense])$
end

инвентаря смыслов I и выбираем смысл, который имеет наибольшее количество признаков смысла с признаками контекста. В случае если общих признаков не найдено, используется смысл, соответствующий наибольшему кластеру (как правило, представляющему доминантный смысл слова).

Алгоритм начинает работу с извлечения индуцированных смысловых кластеров целевого слова (строка 1). Затем метод накапливает веса контекстных признаков каждого смысла *sense* в $\alpha[sense]$. Каждое слово w в кластере смысла *cluster* содержит все свои контекстные признаки $F(w, c)$: см. строки 5–12. Наконец, выбирается смысл *sense*, который максимизирует средний вес по контекстным признакам относительно признаков смыслов из инвентаря (строки 13–21). Если ни один из контекстных признаков не соответствует смысловым представлениям используется наибольший кластер (если задана опция LCB).

4.3 Результаты

Для экспериментов используются две стандартные коллекции, подходящие для оценки методов дизамбигуации: набор данных WSI [56] и SemEval-2013 [57]. Представленный метод показал качество дизамбигуации, сравнимую с другими методами без учителя в том числе нейронных. Однако ни одна из конкурирующих систем не имеет сопоставимого с нашим подходом уровня интерпретируемости результатов.

Детали экспериментальных результатов и их анализ можно найти в [1, 18].

В дополнение к экспериментальным результатам была выпущена реализация метода с открытым исходным кодом, включающая веб-демонстрацию нескольких предварительно обученных моделей.[1] Архитектура системы включает API и веб-приложение с пользовательским интерфейсом для интерпретируемой системой дизамбигуации и навигации по инвентарю смыслов. Приложение выполняет интерпретируемое для пользователя устранение неоднозначности слов в тексте, введенного пользователем. Обладает режимом устранения неоднозначности одного слова (см. Рисунок 4-3) и в режиме устранения неоднозначности всех слов в тексте одновременно (см. Рисунок 4-4).

[1] `http://www.jobimtext.org/wsd`

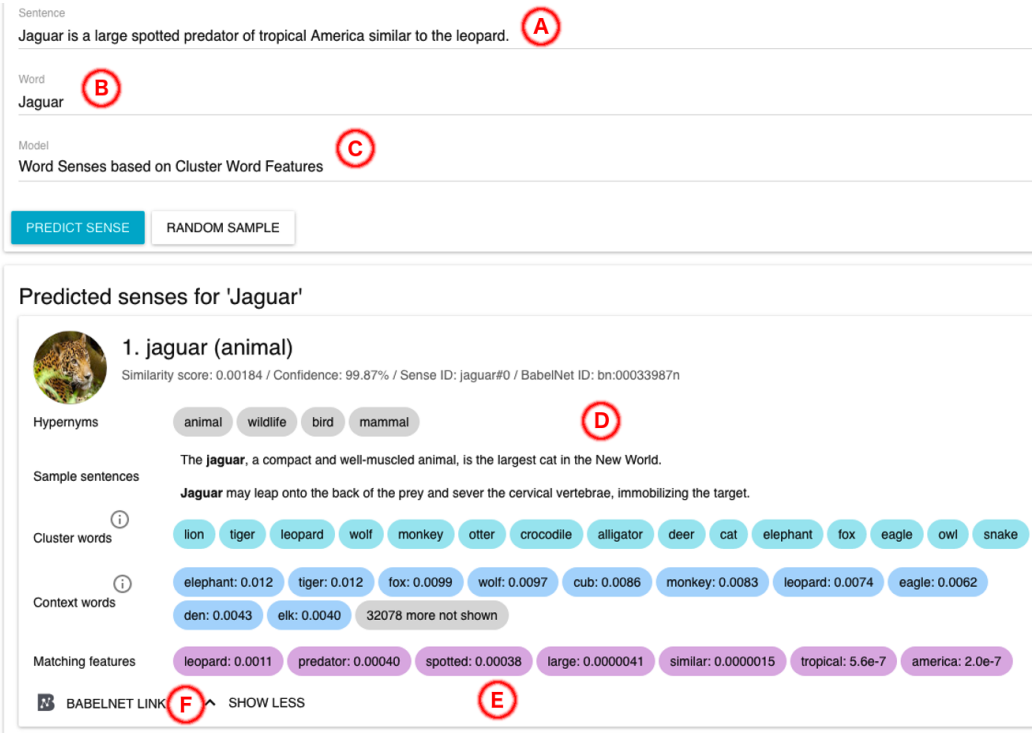

Рис. 4-3: Режим устранения неоднозначности по одному слову: устранение неоднозначности слова “Jaguar” (B) в предложении “*Jaguar* is a large spotted predator of tropical America similar to the leopard.” Предсказанный смысл суммируется с помощью гиперонима и изображения (D) и дополнительно представлен примерами использования, семантически связанными словами и типичными контекстными подсказками. Каждый из этих элементов извлекается автоматически. Смыслы слов связаны с BabelNet (F).

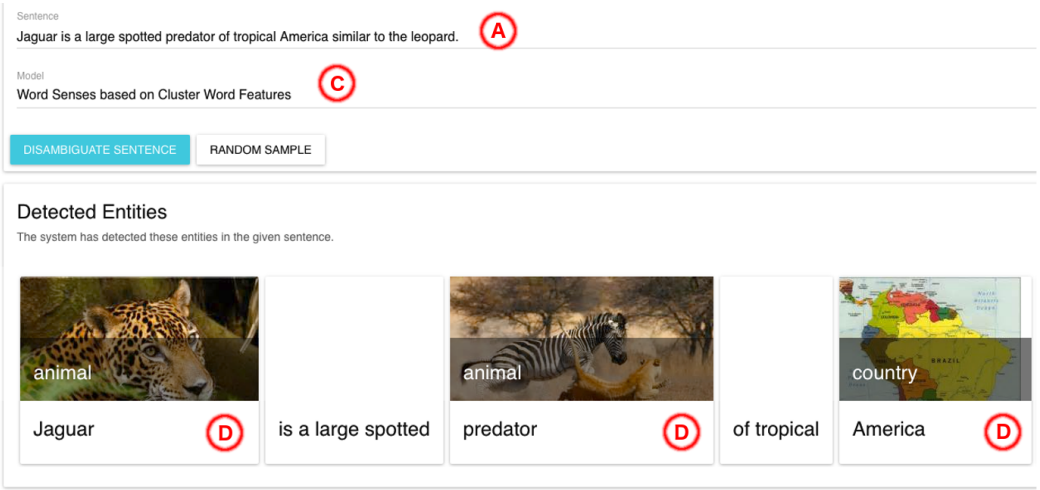

Рис. 4-4: Режим разрешения неоднозначности всех слов: результаты дизамбигуации существительных.

Глава 5

Связывание представлений смыслов

Материалы данной главы основаны на статьях [7,8]из списка 14 публикаций на которых основана диссертация.

5.1 Введение

В этой главе рассмотрены методы связывания смысловых представлений слов, индуцированных из текста методами, представленными в предыдущих главах, с лексико-семантическими сетями, построенными вручную, такими как WordNet. Связь осуществляется на уровне отдельных смыслов слов.

Задача **связывания смыслов слов** заключается в следующем. Входные данные: (i) **построенный вручную** лексико-семантический граф W, напр. WordNet и (ii) **дистрибутивный** лексико-семантический граф, такой граф используемый в предыдущей главе, $T = \{(j, R_j, H_j)\}$, где j — идентификатор смысла, например mouse:1, R_j набор его семантически близких смыслов, например $\{\text{keyboard:1}, \text{computer:0}, \dots\}$, H_j множество гиперонимов, например $\{\text{equipment:3}, \dots\}$. Выходными данными является **соответствие** M: набор пар, например $(source, target)$, где $source \in T.senses$ — это смысл T, а $target \in W.senses \cup source$ — наиболее подходящий смысл W.

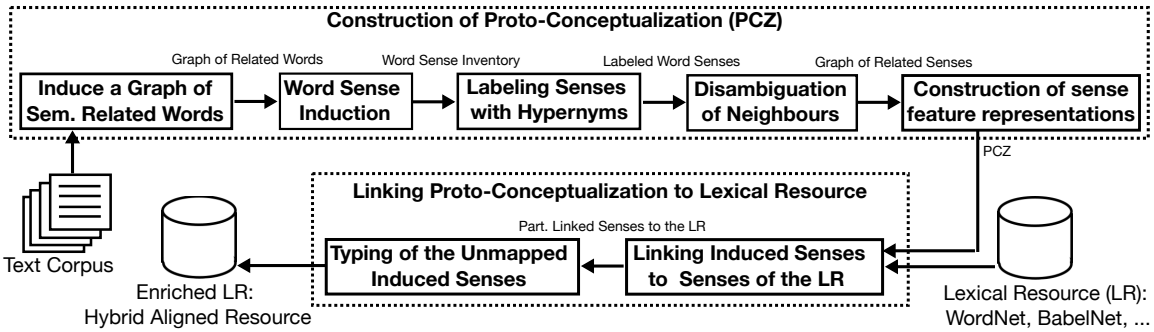

Рис. 5-1: Обзор предложенного подхода для обогащения лексических ресурсов: дистрибутивная модель используется для построения однозначной дистрибутивной лексико-семантической сети (proto-conceptualization, PCZ), которая впоследствии связывается с лексическим ресурсом.

5.2 Метод

Создание гибридного выровненного ресурса (hybrid aligned resource, HAR) основано на методах, используемых для связывания лексических ресурсов, таких как BabelNet [58] и UBY [59]. Однако в данном случае связывание осуществляется между двумя ресурсами, которые были построены совершенно разными способами: ручным и автоматическим.

Связывание индуцированных смыслов со смыслами лексического ресурса

Каждое значение из PCZ связывается с наиболее подходящим значением лексического ресурса (LR) если таковое имеется , см. Рис. 5-1, шаг 3). Существует множество алгоритмов связывания баз знаний [60].Например, [59, 61] производят вычисление перекрытия между наборами слов, построенными на основе смыслов из LR. Предложенный метод использует итеративный подход для того чтобы связывание могло извлечь выгоду из наличия связанных смыслов из предыдущих итераций.Алгоритм 4 принимает на вход:

1. а PCZ $T = \{(j_i, R_{j_i}, H_{j_i})\}$ где j_i — идентификатор смысла (например, mouse:1), R_{j_i} набор семантически связанных смыслов (например, $R_{j_i} = \{$keyboard:1, computer:0, $\dots\}$ и H_{j_i} набор гипернимов (например, $H_{j_i} = \{$equipment:3, $\dots\}$);

2. лексический ресурс W такой как, WordNet или BabelNet;

3. порог th по сходству между парами смыслов и число m итераций как критерий

Algorithm 4 Связывание индуцированных смыслов слов с лексическим ресурсом.

Input: $T = \{(j_i, R_{j_i}, H_{j_i})\}$, W, th, m
Output: $M = (source, target)$
1: $M = \emptyset$
2: **for all** $(j_i, R_{j_i}, H_{j_i}) \in T.monosemousSenses$ **do**
3: $C(j_i) = W.getSenses(j_i.lemma, j_i.POS)$
4: **if** $|C(j_i)| == 1$, let $C(j_i) = \{c_0\}$ **then**
5: **if** $sim(j_i, c_0, \emptyset) \geq th$ **then**
6: $M = M \cup \{(j_i, c_0)\}$
7: **for** $step = 1$; $step \leq m$; $step = step + 1$ **do**
8: $M_{step} = \emptyset$
9: **for all** $(j_i, R_{j_i}, H_{j_i}) \in T.senses/M.senses$ **do**
10: $C(j_i) = W.getSenses(j_i.lemma, j_i.POS)$
11: **for all** $c_k \in C(j_i)$ **do**
12: $rank(c_k) = sim(j_i, c_k, M)$
13: **if** $rank(c_k)$ has a single top value for c_t **then**
14: **if** $rank(c_t) \geq th$ **then**
15: $M_{step} = M_{step} \cup \{(j_i, c_t)\}$
16: $M = M \cup M_{step}$
17: **for all** $(j_i, R_{j_i}, H_{j_i}) \in T.senses/M.senses$ **do**
18: $M = M \cup \{(j_i, j_i)\}$
19: **return** M

остановки.

Алгоритм возвращает соответствие M, которое состоит из набора пар вида $(source, target)$, где $source \in T.senses$ — смысл во входном PCZ T и $target \in W.senses \cup source$ является наиболее подходящим смыслом в лексическом ресурсе W или $source$, если такой смысл не идентифицирован.

Алгоритм начинает работу с создания пустого отображения M (строка 1). Затем для каждого однозначного смысла (например, Einstein:0 — единственное значение в PCZ для термина Einstein) он ищет кандидата в однозначное значение (строки 2-6). Если такие однозначные смыслы-кандидаты существуют (строка 4), мы сравниваем два смысла (строка 5) с помощью функции сходства:

$$sim(j, c, M) = \frac{|T.BoW(j, M, W) \cap W.BoW(c)|}{|T.BoW(j, M, W)|}, \tag{5.1}$$

Then a new link pair (j_i, c_0) is added to M if the similarity score between j_i and c_0 meets or exceeds the threshold th (line 5). At this point, we collected a first set of

disambiguated (monosemous) senses in M and start to iteratively disambiguate the remaining (polysemous) senses in T (lines 7-16). This iterative disambiguation process is similar to the one we described for the monosemous case (lines 2-6), with the main difference that, due to the polysemy of the candidates synsets, we instead use the similarity function to rank all candidate senses (lines 11-12) and select the top-ranked candidates for the mapping (lines 13-15). At the end of each iteration, we add all collected pairs to M (line 16). Finally, all unlinked j of T, i.e. induced senses that have no corresponding LR sense, are added to the mapping M (lines 17- 18).

где

1. $T.BoW(j, M, W)$ — набор слов, содержащий все термины, извлеченные из родственных/гиперонимных значений j, и все термины, извлеченные из родственных/гиперонимных (т.е. уже связанных) в M) смыслов в W. Для каждого смысла из LR мы используем все синонимы и содержательные слова определения.

2. $W.BoW(c)$ содержит синонимы и слова пояснения для смысла c и всех связанных смыслов c.

Затем к M добавляется новая пара соответствий (j_i, c_0), если показатель сходства между j_i и c_0 соответствует или превышает пороговое значение th (строка 5). На этом этапе мы собрали первый набор неоднозначных (моносемных) значений в M и начали итеративно устранять неоднозначность остальных (многозначных) значений в T (строки 7-16). Этот итерационный процесс устранения неоднозначности аналогичен тому, который мы описали для однозначного случая (строки 2-6), с той основной разницей, что из-за многозначности синсетов кандидатов мы вместо этого используйте функцию сходства, чтобы ранжировать все смыслы-кандидаты (строки 11-12) и выбирать кандидатов с самым высоким рейтингом для сопоставления (строки 13-15). В конце каждой итерации мы добавляем все собранные пары в M (строка 16). Наконец, к отображению M (строки 17-18) добавляются все несвязанные j поля T, т.е. индуцированные смыслы, не имеющие соответствующего LR-смысла.

Финальный результат работы описанного метода состоит из: i) прото-концептуализации (PCZ); ii) соответствия M смыслов из PCZ и смыслов лексико-семантического ресурса, такого как WordNet или BabelNet; iii) соответствия

PCZ ID	WordNet ID	PCZ связанные слова	PCZ контекстные слова
mouse:0	mouse:wn1	rat:0, rodent:0, monkey:0, ...	rat:conj_and, gray:amod, ...
mouse:1	mouse:wn4	keyboard:1, computer:0, printer:0 ...	click:-prep_of, click:-nn, ...
keyboard:0	keyboard:wn1	piano:1, synthesizer:2, organ:0 ...	play:-dobj, electric:amod, ..
keyboard:1	keyboard:wn1	keypad:0, mouse:1, screen:1 ...	computer, qwerty:amod ...

Таблица 5.1: Примеры записей гибридного выровненного ресурса (HAR) для слов *mouse* и *keyboard*. Числа обозначают идентификаторы смысла. Чтобы обогатить смысловые представления WordNet мы используем на связанные термины и контекстные признаки.

H предлагаемых типов для записей PCZ, не отображенных в M (алгоритм описан в публикации в приложении).

5.3 Результаты

Оценка качества на основе ручной разметки на разных этапах работы метода, а также внешняя оценка на основе задачи дизамбигуации указывают на высокое качество нового гибридного ресурса (HAR). Кроме того, ресурс был протестирован в задаче построения таксономии.

Примеры связанных смыслов слов представлены в Таблице 5.1 между Wordnet и дистрибутивным тезаурусом, созданным автоматически (PCZ). Дополнительные примеры, детали экспериментальных результатов и их анализ можно найти в [7, 8].

Глава 6

Предсказание векторных представлений гиперонимов

Материал данной главы основаны на статье [4] fиз списка 14 публикаций на которых основана диссертация.

6.1 Введение

В этой главе представлен метод извлечения гиперонимов, основанный на проекции векторных представлений слов.

Гипернимия — это иерархическое семантическое отношение между словами, такое как, (apple, fruit) или (jaguar, animal). В первом примере слово apple является **гипонимом**, а слово fruit — **гипернимом**. Задаче, которую мы рассматриваем в этой главе – предсказание для заданного гипонима его гипернима.

В отличие от подходов, основанных на классификации, методы, основанные на проекциях, не требуют пар-кандидатов гипоним-гиперним. Однако при извлечении отношений с учителем естественно использовать как положительные, так и отрицательные обучающие примеры, влияние отрицательных примеров на гипернимное предсказание ранее не изучалось. В этой главе показано, что явные отрицательные примеры, используемые для регуляризации модели, значительно повышают производительность по сравнению с базовым подходом для обучения проекции [62] на трех наборах данных для различных языков.

6.2 Метод

Предложенный метод выполняет извлечение гипернимии посредством регуляризованного обучения проекциям.

Базовый подход В качестве базового используется модель [62]. В этом подходе матрица проекции $\mathbf{\Phi}^*$ получается аналогично задаче линейной регрессии, т.е. для заданных векторов-строк слов $\vec{x}$ и $\vec{y}$, представляющих соответственно гипоним и гипероним, квадратная матрица $\mathbf{\Phi}^*$ подходит на обучающем наборе положительных пар $\mathcal{P}$:

$$\mathbf{\Phi}^* = \arg\min_{\mathbf{\Phi}} \frac{1}{|\mathcal{P}|} \sum_{(\vec{x},\vec{y})\in\mathcal{P}} \|\vec{x}\mathbf{\Phi} - \vec{y}\|^2, \tag{6.1}$$

где $|\mathcal{P}|$ — количество обучающих примеров, а $\|\vec{x}\mathbf{\Phi} - \vec{y}\|$ — расстояние между парой векторов-строк $\vec{x}\mathbf{\Phi}$ и $\vec{y}$. В исходном методе используется расстояние L^2 . Для повышения производительности k матрицы проекций $\mathbf{\Phi}$ изучаются по одной для каждого кластера отношений в обучающем наборе. Один из примеров представлен смещением гипонима-гипернима. Кластеризация выполняется с использованием алгоритма k-средних [63].

Лингвистические ограничения посредством регуляризации Ближайшие соседи, созданные с использованием векторов векторных представлений слов, обычно содержат смесь синонимов, гиперонимов, когипонимов и других родственных слов [64–66]. Чтобы явно предоставить примеры нежелательных отношений в модели, мы предлагаем две улучшенные версии базовой модели: *асимметричная регуляризация*, которая использует инвертированные отношения в качестве отрицательных примеров, и *регуляризация соседей*, которая использует отношения других типов. как отрицательные примеры. Для этого мы добавляем член регуляризации к функции потерь:

$$\mathbf{\Phi}^* = \arg\min_{\mathbf{\Phi}} \frac{1}{|\mathcal{P}|} \sum_{(\vec{x},\vec{y})\in\mathcal{P}} \|\vec{x}\mathbf{\Phi} - \vec{y}\|^2 + \lambda R, \tag{6.2}$$

где λ — константа, контролирующая важность члена регуляризации R.

Асимметричная регуляризация. Поскольку гипернимия — это асимметричное отношение, наш первый метод обеспечивает асимметрию матрицы проекции.

Применение того же преобразования к предсказанному вектору гиперонима $\vec{x}\mathbf{\Phi}$ не должно привести к получению вектора, подобного $(\cdot)$ исходному вектору гипонима $\vec{x}$. Обратите внимание, что этот регуляризатор требует только положительных примеров $\mathcal{P}$:

$$R = \frac{1}{|\mathcal{P}|} \sum_{(\vec{x},\_)\in\mathcal{P}} (\vec{x}\mathbf{\Phi}\mathbf{\Phi} \cdot \vec{x})^2. \tag{6.3}$$

Регуляризация с использованием ближайших соседей. Этот подход основан на отрицательной выборке путем явного предоставления примеров семантически связанных слов $\vec{z}$ гипонима $\vec{x}$, которые наказывают матрицу для создания векторов, подобных их:

$$R = \frac{1}{|\mathcal{N}|} \sum_{(\vec{x},\vec{z})\in\mathcal{N}} (\vec{x}\mathbf{\Phi}\mathbf{\Phi} \cdot \vec{z})^2. \tag{6.4}$$

Этот регуляризатор требует отрицательных примеров $\mathcal{N}$. В данном случае были использованы синонимы гипонимы $\mathcal{N}$, однако можно использовать и другие типы отношений, такие как антонимы, меронимы или когипонимы. Некоторые слова могут не иметь синонимов в обучающем наборе. В таких случаях мы заменяем $\vec{z}$ на $\vec{x}$, откатываясь к предыдущему варианту модификации базового подхода. В противном случае в каждую эпоху обучения мы выбираем случайный синоним данного слова.

Регуляризаторы без повторного проецирования. Помимо двух описанных выше регуляризаторов, основанных на перепроецировании вектора гипонима $(\vec{x}\mathbf{\Phi}\mathbf{\Phi})$, мы также протестировали два регуляризатора без перепроектирования, обозначаемые как $\vec{x}\mathbf{\Phi}$. Регуляризатор без повторного использования семантических близайших соседей определяется следующим образом:

$$R = \frac{1}{|\mathcal{N}|} \sum_{(\vec{x},\vec{z})\in\mathcal{N}} (\vec{x}\mathbf{\Phi} \cdot \vec{z})^2. \tag{6.5}$$

В данном случае регуляризатор пенализирует связанность предсказанного гипернима $\vec{x}\mathbf{\Phi}$ с синонимом $\vec{z}$. Аналогично определяется асимметричный регуляризатор без перепроецирования.

Обучение моделей Для обучения моделей используется метод оптимизации Адам [67] с метапараметрами по умолчанию, реализованными в TensorFlow [68]. Было

проведено 700 эпох обучения, где на каждой эпохе подавалась группа из 1024 обучающих примеров. Матрицы проекций инициализировались с использованием нормального распределения $\mathcal{N}(0, 0.1)$.

6.3 Результаты

Оценка предложенных методов проводилась на одном русском и двух английских наборах данных по гипернимии. Эксперименты в контексте задачи прогнозирования гипернимии для обоих языков показывают значительные улучшения предлагаемого подхода по сравнению с современными аналогами.

Детали экспериментальных результатов и их анализ можно найти в [4].

Глава 7

Извлечение гиперонимов с помощью кластеризации смыслов

Материал данной главы основаны на статье [19] из списка 14 публикаций на которых основана диссертация.

7.1 Введение

В этой главе показано, как семантические классы слов построенные автоматически могут быть полезны для извлечения гипернимов.

Задача, которая рассматривается в данной главе заключается в следующем. Имея множество **зашумленных гиперонимов** $H = \{(w_i, w_j), (w_k, w_l), ..., (w_y, w_z)\}$ получить обновленное множество **очищенных гиперонимов** H', которое (i) не будет содержать неверных отношений, (ii) будет содержать отсутствующие отношения.

Ниже представлены методы создания семантических классов с использованием методов кластеризации дистрибутивных графов смыслов слов. Семантические классы используются для фильтрации зашумленных отношений гипернимии. Очистка гипернимов осуществляется путем автоматической разметки каждого семантического класса его гиперонимами. С одной стороны, это позволяет нам отфильтровывать неправильные отношения, используя глобальную структуру распределения схожих смыслов. С другой стороны, мы делаем вывод об отсутствующих гиперонимах посредством распространения меток класса на термины кластера слов. Мы проводим

крупномасштабное краудсорсинговое исследование, показывающее, что обработка автоматически извлеченных гипернимов с использованием нашего подхода улучшает качество извлечения гипернимии с точки зрения как точности, так и полноты. Кроме того, полезность метода продемонстрирована в задаче индукции таксономии предметной области в задаче SemEval-2016.

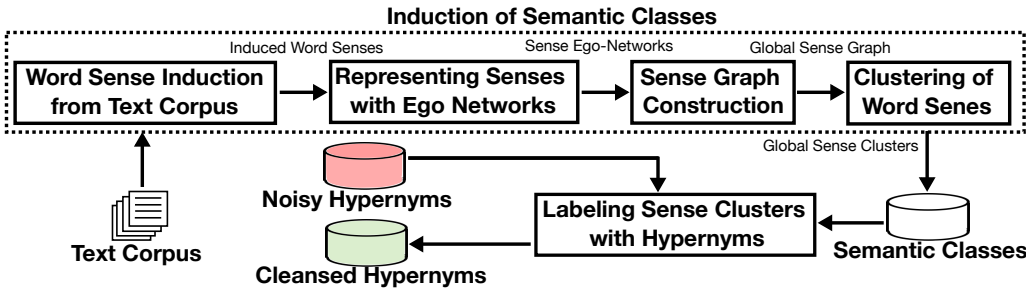

Рис. 7-1: Краткое описание подхода: семантические классы извлекаются из текстового корпуса, а затем используются для фильтрации базы данных зашумленных гипернимов (например, извлеченных внешним методом из текстового корпуса).

7.2 Метод

В этом разделе представлен метод индукции семантических классов и метод извлечения гиперонимов, используя индуцированные классы. Как показано на Рис. 7-1, метод проводит извлечение смыслов из корпуса текста, используя метод описанный выше и в [69, 70], и группирует глобально полученные смыслы.

Примеры значений слов из индуцированных семантических кластеров представлены в Таблице 7.1. Семантические классы представляют собой глобальную, а не локальную кластеризацию смыслов, т.е. слово “apple” в смысле “fruit” может быть членом только одного кластера. Ниже описывается каждый шаг предложенного метода.

Индукция смысла слова по текстовому корпусу

Каждый смысл слова s в индуцированном смысловом инвентаре $\mathcal{S}$ представлен списком соседей $\mathcal{N}(s)$. Извлечение этой сети выполняется с использованием метода [69] и включает в себя три этапа: (1) построение дистрибутивного тезауруса, т.е. графа связанных неоднозначных терминов [71]; (2) индукция смысла слова

Глобальный кластер смыслов: семантический класс, $c \subset \mathcal{S}$	**Гиперонимы,** $\mathcal{H}(c) \subset \mathcal{S}$
peach#1, banana#1, pineapple#0, berry#0, blackberry#0, grapefruit#0, strawberry#0, blueberry#0, fruit#0, grape#0, melon#0, orange#0, pear#0, plum#0, raspberry#0, watermelon#0, apple#0, apricot#0, watermelon#0, pumpkin#0, berry#0, **mangosteen#0**, ...	vegetable#0, fruit#0, crop#0, ingredient#0, food#0, ·
C#4, Basic#2, Haskell#5, Flash#1, Java#1, Pascal#0, Ruby#6, PHP#0, Ada#1, Oracle#3, Python#3, Apache#3, Visual Basic#1, ASP#2, Delphi#2, SQL Server#0, CSS#0, AJAX#0, JavaScript#0, SQL Server#0, Apache#3, Delphi#2, Haskell#5, .NET#1, CSS#0, ...	programming language#3, technology#0, language#0, format#2, app#0

Таблица 7.1: Примеры извлеченных кластеров смыслов, представляющие семантические классы “fruits” и “programming language”.

посредством кластеризации эго-сетей [72, 73] связанных слов с использованием алгоритма кластеризации графов [45]; (3) устранение неоднозначности родственных слов и гиперонимов.

Senses in the induced sense inventory may contain a mixture of different senses introducing noise in a global clustering, e.g. “Python” in the animal sense is related to both car and snake senses. To minimize the impact of the word sense induction errors, we filter out ego networks with a highly segmented structure. Namely, we cluster each ego network with the Chinese Whispers algorithm and discard networks for which the cluster containing the target sense s contains less than 80% nodes of the respective network to ensure semantic coherence inside the word groups. Besides, all nodes of a network not appearing in the cluster containing the ego sense s are also discarded.

Представление смыслов слов с помощью эго сетей

Чтобы выполнить глобальную кластеризацию извлеченных смыслов, мы представляем каждый смысл s с помощью *эго-сети* [73] второго порядка. Эго-сеть — это граф, состоящий из всех связанных смыслов $\mathcal{R}(s)$ смысла s, достижимых по пути длиной один или два, определяемых как:

$$\{s_j : (s_j \in \mathcal{N}(s)) \vee (s_i \in \mathcal{N}(s) \wedge s_j \in \mathcal{N}(s_i))\}. \tag{7.1}$$

Каждый вес ребра $\mathcal{W}_s(s_i, s_j)$ между двумя смыслами задается равен значению семантической близости распределения между s_i и s_j.

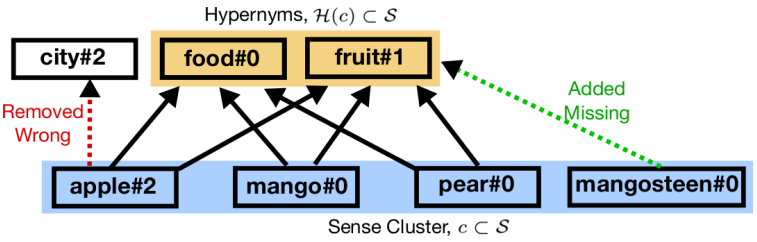

Рис. 7-2: Предложенный подход выполняет обнаружение ошибочных и генерирование недостающих гипернимов с использованием семантических классов размеченных гиперонимами.

Смыслы в инвентаре могут содержать смесь различных смыслов, вносящих шум в глобальную кластеризацию.Чтобы свести к минимуму влияние ошибок индукции смысла слова, мы отфильтровываем эго-сети с сильно сегментированной структурой. А именно, мы кластеризуем каждую эго-сеть с помощью алгоритма китайского шепота и отбрасываем сети, для которых кластер, содержащий целевой смысл s, содержит менее 80% узлов соответствующей сети, чтобы обеспечить семантическую согласованность внутри групп слов.

Построение глобального графа смыслов

Цель этого шага — объединить эго-сети отдельных смыслов, построенные на предыдущем этапе, в глобальный граф. Мы вычисляем веса ребер глобального графа подсчитывая количество совпадений одного и того же ребра в разных эго-сетях:

$$\mathcal{W}(s_i, s_j) = \sum_{s \in \mathcal{S}} \mathcal{W}_s(s_i, s_j). \tag{7.2}$$

Для фильтрации нерелевантных ребер удаляются ребра с весом меньше заданного порога t. После чего , мы применяем функцию $E(w)$, которая масштабирует веса ребер:

$$\mathcal{W}(s_i, s_j) = \begin{cases} E(\mathcal{W}(s_i, s_j)) & \text{if } \mathcal{W}(s_i, s_j) \geq t, \\ 0 & \text{otherwise.} \end{cases} \tag{7.3}$$

Кластеризация смыслов слов

Ядром метода является индукция семантических классов путем кластеризации глобального графа значений слов. Используется алгоритм жесткой кластеризации графов Chinese Whispers, для того чтобы каждый смысл оказался в одном кластере c. Результатом работы алгоритма являются группы семантически близких значений

слов. Были произведены две версии кластеризации: *гранулированная* группирует 208871 индуцированных смыслов в 1870 семантических классов, и *грубая*, которая группирует 18028 смыслов слов в 734 семантических класса. Чтобы найти оптимальные параметры метода, индуцированные кластеры сравнивались с лексико-семантическими кластерами из WordNet 3.1 [74] и BabelNet 3.7 [75].

Извлечение гипернимов с использованием семантических классов

Размечая семантические классы гипернимами возможно удалить неправильные или добавить недостающие отношения, как показано на Рис. 7-2. Каждый кластер размечен гипернимами, где метки представляют собой общие гипернимы слов в кластере (см. Таблицу 7.1). Новые гиперонимы получаются путем распространения меток кластера на редкие слова без гипернимов, например “mangosteen” на Рис. 7-2. Гипернимы, которые встречаются во многих смыслах s, имеют пониженный вес, который вычисляется следующим образом:

$$\text{tf–idf}(h) = \sum_{s \in c} \mathcal{H}(s) \cdot \log \frac{|\mathcal{S}|}{|h \in \mathcal{H}(s) : \forall s \in \mathcal{S}|}, \tag{7.4}$$

где $\sum_{s \in c} \mathcal{H}(s)$ — сумма весов всех гипернимов для каждого смысла s для каждого кластера c. Мы размечаем каждый кластер c пятью гиперонимами $\mathcal{H}(c)$.

7.3 Результаты

Для оценки подхода были проведены три эксперимента. Крупномасштабное краудсорсинговое исследование показало высокую достоверность извлеченных семантических классов по мнению человека. Кроме того, было продемонстрировано, что этот подход помогает повысить точность и запоминаемость метода извлечения гипернимии. Наконец, было показано, что семантические классы можно использовать для улучшения индукции таксономии предметной области из текста (на основе набора данных SemEval-2016).

Детали экспериментальных результатов и их анализ можно найти в [19].

Глава 8

Построение таксономий с помощью гиперболических векторов

Материалы данной главы основаны на статье [5] из списка 14 публикаций на которых основана диссертация.

8.1 Введение

Задача достроения таксономии, рассмотренная в данной главе, состоит в следующем. Задан **неполный таксономический граф** $G = (V, E)$, где ребра E представляют собой отношения гипернимии между словами в V с ошибками двух типов (i) отсутствующие ребра: $E_{abs} = \{(v_i, v_j) : v_i \text{ is-a } v_j \wedge (v_i, v_j) \notin E\}$ с особым случаем незакрепленных узлов $V_{orh} = \{v_i \in V : \nexists (v_i, v_j) \in E\}$; и (ii) неправильные ребра: $E_{wrg} = \{(v_i, v_j) \in E : v_i \text{ not-is-a } v_j\}$ построить **полный таксономический граф** $G' = (V, E')$ исправляя ошибки путем: (i) добавления отсутствующих ребер: $E = E \cup \{E_{abs}\}$; (ii) удаления неправильных ребер: $E = E \setminus E_{wrg}$. Для незакрепленных узлов требуется добавление ребер для того чтобы сделать граф связным. Для связанных узлов необходимо либо добавить отсутствующее дополнительное ребро, либо переместить, если оно размещено неправильно. Последний представляет собой комбинацию удаления неправильного ребра и добавления отсутствующего ребра к рассматриваемому отсоединенному узлу.

В этой главе представлен метод использования гиперболических векторных

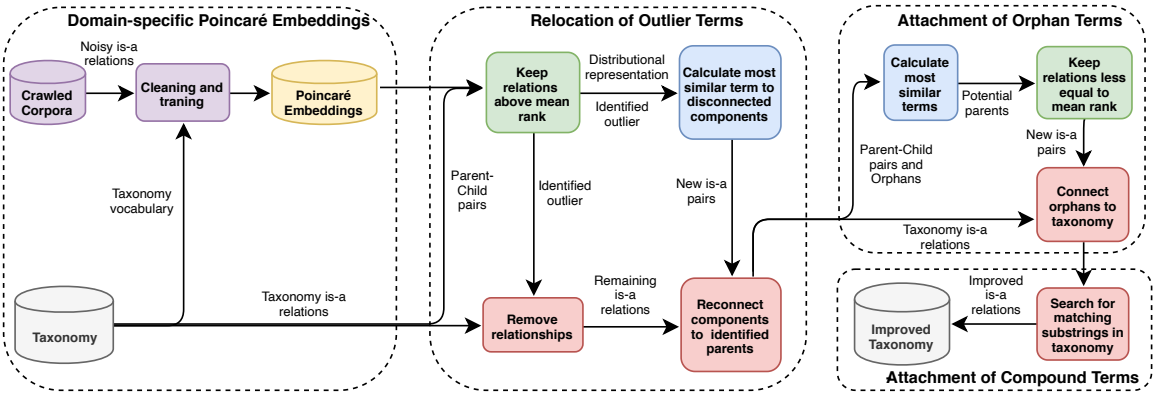

Рис. 8-1: Схема метода достроения таксономии.

представлений слов для достроения таксономии предметной области. Метод существенно улучшает предыдущие результаты по достроению таксономии. Мы демонстрируем превосходство гиперболических эмбеддингов над стандартными эмбеддингами показывая, что они могут лучше отражать иерархические лексико-семантические отношения, чем эмбеддинги в евклидовом пространстве.

8.2 Метод

В этом разделе представлен метод достроения таксономии с использованием гиперболических векторных представлений слов (см. Рис. 8-1). Эмбеддинги на основе дистрибутивной семантики (такие как, word2vec) и гиперболические эмбеддинги [76] используются для устранения ошибок в таксономии. Первый шаг метода заключается в создании гиперболических эмбеддингов для требуемой предметной области. Затем полученные эмбеддинги используются для идентификации и перемещения неверных отношений в таксономии, а также для присоединения несвязанных терминов. На последнем шаге мы дополнительно оптимизируем таксономию, используя удаляя циклы.

Построение обучающего набора данных

Для создания гиперболических эмбеддингов, специфичных для предметной области, мы используем отношения гипернимии извлеченные из комбинации общих и специфичных для предметной области корпусов текстов. Для общего домена мы извлекли текст из английской Википедии, корпуса Gigaword [77], корпуса ukWac [78] и лейпцигского новостного корпуса [79]. Отношения гиперонимии извлекаются с

помощью лексико-синтаксических шаблонов из всех корпусов путем применения библиотек PattaMaika, PatternSim [80] и WebISA [81] как было сделано в [82].

Извлеченные отношения общих и предметно-специфичных корпусов объединяются. Чтобы ограничить количество терминов и отношений, мы ограничиваем отношения парами, для которых оба слова являются частью словаря таксономии. Отношения с частотой менее трех удаляются для фильтрации неверных извлечений. Для всех рефлексивных отношений, сохраняется только наиболее часто встречающееся отношение. В следствие этого, извлеченные отношения являются антисимметричными и иррефлексивными.

Та же процедура применяется к отношениям, извлеченным из корпуса общего домена. Затем они используются для расширения набора отношений, созданных на основе корпусов, специфичных для предметной области.

Расстояние между гипонимами и гиперонимами

Гиперболические эмбеддинги обучаются на извлеченных доменно-специфичных отношениях. Для сравнения результатов, в дополнение была обучена модель на парах существительных извлеченных из WordNet. Кроме этого, были обучены дистрибутивные эмбеддинги таким образом, что словосочетания из словаря таксономии в обучающем корпусе были представлены отдельным вектором.

В отличие от эмбеддингов в евклидовом пространство, где косинусное подобие обычно применяется как мера семантического расстояния, т.е.

$$d(\mathbf{u}, \mathbf{v}) = 1 - \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}||\mathbf{v}|}, \tag{8.1}$$

Гиперболические векторные представления используют гиперболическое пространство, в частности модель шара Пуанкаре [83]. Гиперболические эмбеддинги хорошо приспособлены для моделирования иерархических отношений между словами, поскольку они явно отражают иерархию между словами в векторном пространстве. Расстояние между двумя точками $\mathbf{u}, \mathbf{v} \in \mathcal{B}^d$ для d-мерной модели шара Пуанкаре определяется как, известное как расстояние Пуанкаре:

$$d(\mathbf{u}, \mathbf{v}) = \operatorname{arcosh}\Big(1 + 2\frac{||\mathbf{u} - \mathbf{v}||^2}{(1 - ||\mathbf{u}||^2)(1 - ||\mathbf{v}||^2)}\Big). \tag{8.2}$$

Расстояние Пуанкаре позволяет нам одновременно оценить иерархию и

семантическую близость между словами. Оно увеличивается экспоненциально с глубиной семантической иерархии. Таким образом, хотя расстояние от листового узла до большинства других узлов иерархии очень велико, расстояние от корня и узлов на высоких уровнях сравнительно невелико до всех узлов иерархии. С другой стороны, дистрибутивные векторные представления моделируют, как правило, не гиперонимию, а когипонимию: близкие слова в подобных евклидовых пространствах часто являются когипонимами [84, 85].

Исправление неверных отношений

Гиперболические эмбеддинги используются для вычисления ранга $rank(x, y)$ между каждым дочерним и родительским термином в исходной таксономии, определяемого как индекс y в списке отсортированных расстояний Пуанкаре всех объектов таксономии до x. Из таксономии удаляются отношения с рангом большим среднего всех рангов, выбранных на основе тестовых данных TExEval [86]. В случае если отсоединенный узел имеет дочерние узлы они прикрепляются к наиболее похожему родительскому узлу в таксономии либо к корню таксономии.

Поиск родителей для несвязанных узлов

С помощью дистрибутивных эмбеддингов вычисляется расстояние до ближайшего когипонима для каждого несвязанного узла. Далее с помощью гиперболических эмбеддингов вычисляется ранг между каждым таким кандидатом и остальными узлами таксономии. В таксономию добавляются отношения гипероним-гипоним с рангом ниже или равным среднему значению всех сохраненных рангов. Таким образом, устанавливается связь между родителем наиболее похожего когипонима и несвязным узлом.

В упрощенном методе, без использования гиперболических эмбеддингов, отношение устанавливается между родителем когипонима, полученного из таксономии.

Поиск родителей для слососочетаний

Если вектор составного термина не найден, мы представляем его основным словом в данном словосочетании. В заключение, используется алгоритм Тарьяна [87], чтобы гарантировать асимметричность уточненной таксономии. В случае обнаружения цикла в графе одна из ребер цикла удаляется случайным образом.

Слово	**Родитель: паттерны**	**Родитель: гиперболические эмбеддинги**	**Верный родитель**	**Ближайшие соседи**
second language acquisition	—	linguistics	linguistics	applied linguistics, semantics, linguistics
botany	—	genetics	plant science, ecology	genetics, evolutionary ecology, animal science
sweet potatoes	—	vegetables	vegetables	vegetables, side dishes, fruit
wastewater	water	waste	waste	marine pollution, waste, pollutant
water	waste, natural resources	natural resources	aquatic environment	continental shelf, management of resources
international relations	sociology, analysis, humanities	humanities	political science	economics, economic theory, geography

Таблица 8.1: Примеры слов с соответствующими родительскими элементами во входной таксономии, построенными с использованием шаблонов и после уточнения с использованием гиперболических эмбеддингов.

8.3 Результаты

Оценка предлагаемого метода была выполнена на наборе данных уточнения таксономии SemEval-2016 и на базе трех лучших систем. Примеры предсказаний приведены в Таблице 8.1. Эксперименты показывают, что разработанный метод уточнения таксономий за счет использования гиперболических эмбеддингов дает более хорошие результаты по сравнению методами основанными на векторных представлениях евклидовом пространстве. Было показано, что гиперболические векторные представления могут быть эффективно построены для заданного домена из текста без необходимости использования существующей базы данных, такой как WordNet. Это наблюдение подтверждает теоретическую способность гиперболических векторных представлений моделировать иерархические отношения между словами, что позволяет в будущем использовать их в широком круге семантических задач.

Детали экспериментальных результатов и их анализ можно найти в [5].

Глава 9

Векторные представления узлов лексико-семантических графов

Материал данной главы основаны на статье [3] из списка 14 публикаций на которых основана диссертация.

9.1 Введение

В этой главе представлены методы обучения векторных представлений узлов лексико-семантических графов для вычисления графовых метрик, такими как кратчайшее расстояние между узлами.

Задача **обучения графовой метрики** состоит в следующем. Для заданного графа $G = (E, V)$ и **графовой метрики** $sim : E \times E \to [0; 1]$ требуется найти матрицу **эмбеддингов узлов** $\mathbf{E} \in R^{|E| \times d}$, где d — размерность эмбеддинги, такую, что $sim(e_i, e_j) \approx f(\mathbf{e}_i, \mathbf{e}_j)$, где f — некоторое векторное расстояние, вычисляемое значительно быстрее, чем sim.

При работе с большими графами, такими как транспортные сети, социальные сети или лексическико-семантические ресурсы, часто возникает необходимость оценки близости между узлами. Во многих предметных областях и для конкретных приложений предложены специальные метрики сходства узлов графа $sim : V \times V \to \mathbb{R}$ были определены на парах узлов V графа $G = (V, E)$. Примеры включают определения времени в пути, алгоритмы детекции сообществ или

семантические расстояния на базе WordNet [88] в алгоритмах дизамбигуации. Например, сходство s_{ij} между синсетами cup.n.01 и mug.n.01 в WordNet равно $\frac{1}{4}$ согласно метрике инвертированного расстояния кратчайшего пути, поскольку эти два узла соединены ненаправленным путем cup $\rightarrow$ container $\leftarrow$ vessel $leftarrow$ drinking_vessel $\leftarrow$ mug.

В литературе описано большое количество подобных метрик сходства узлов графа, многие из которых основаны на алгоритме случайного блуждания в графе [89–91]. В частности, большинство метрик производят итеративный обход ребер графа E, что делает их вычисление чрезмерно неэффективным.

Предложенная модель *path2vec* решает эту проблему за счет разделения этапов на вычислительно сложный этап обучения эмбеддингов и вычислительно эффективный этап вычисления метрик при использовании. Узлы графа представляются эмбеддингами, за счет этого операции в векторном пространстве выполняются на несколько порядков быстрее, чем вычисления метрик непосредственно на графе.

9.2 Метод

Определение модели

Path2vec — модель для обучения графовой метрики, которая строит векторные представления узлов графа $\{v_i, v_j\} \in V$ такие, что скалярные произведения между парами соответствующих векторов $(\mathbf{v}_i \cdot \mathbf{v}_j)$ близки к заданным пользователем мерой близости между узлами s_{ij}. Кроме того, модель усиливает сходство $\mathbf{v}_i \cdot \mathbf{v}_n$ и $\mathbf{v}_j \cdot \mathbf{v}_m$ между узлами v_i и v_j и всеми соответствующими им смежными узлами $\{v_n : \exists(v_i, v_n) \in E\}$ и $\{v_m : \exists(v_j, v_m) \in E\}$, чтобы более точно представить локальную структуру графа. Модель учитывает как глобальные так и локальные отношения между узлами путем минимизации

$$\mathcal{L} = \sum_{(v_i, v_j) \in B} ((\mathbf{v}_i^\top \mathbf{v}_j - s_{ij})^2 - \alpha(\mathbf{v}_i^\top \mathbf{v}_n + \mathbf{v}_j^\top \mathbf{v}_m)), \tag{9.1}$$

где $s_{ij} = sim(v_i, v_j)$ — значение целевой метрики близости между парой узлов v_i и v_j, $\mathbf{v}_i$ и $\mathbf{v}_j$ — эмбеддинги первого и второго узла, B — набор обучающих примеров, α — коэффициент регуляризации. Второй член $(\mathbf{v}_i \cdot \mathbf{v}_n + \mathbf{v}_j \cdot \mathbf{v}_m)$ в целевой функции — это

регуляризатор, который помогает модели одновременно максимизировать сходство между соседними узлами, одновременно максимизируя подобие между смежными узлами (смежный узел выбирается случайным образом для каждого целевого узла).

Мы используем отрицательные обучающие примеры в обучающем наборе B, добавляя p отрицательных примеров ($s_{ij} = 0$) для каждого реального ($s_{ij} > 0$) обучающего примера. В частности, для каждого реального узла паре (v_i, v_j) с подобием s_{ij} сопутствуют p "отрицательные" пары узлов (v_i, v_k) и (v_j, v_l) с нулевым подобием, где v_k и v_l — это случайно выбранные узлы из V. Эмбеддинги инициализируются случайным образом и обучаются с помощью оптимизатора *Adam* [92] с ранней остановкой. После обучения модели вычисление сходства узлов аппроксимируется скалярным произведением обученных векторов узлов, что делает вычисления эффективными: $\hat{s}_{ij} = \mathbf{v}_i \cdot \mathbf{v}_j$.

Связь с подобными моделями

Предложенная модель имеет общие черты с моделю Skip-gram [93], в которой скалярное произведение $\mathbf{v}_i \cdot \tilde{\mathbf{v}}_j$ векторов пар слов (v_i, v_j) из обучающего корпуса должно стремиться к 1 для положительных примеров, в то время как скалярные произведения отрицательных примеров должны быть близки к 0. В модели Skip-gram цель обучения заключается в минимизации условных вероятностей наблюдения слова из контекста w_j при условии текущего слова w_i:

где B_p — набор положительных обучающих примеров, B_n — набор сгенерированных отрицательных примеров, а σ — сигмоида. При этом, Skip-gram использует только локальную информацию, не создавая матрицу совместных встречаемости слов. В предложенной модели *path2vec* целевые значения скалярного произведения s_{ij} не являются бинарными, а могут принимать произвольные значения в диапазоне [0...1], в соответствии с заданной пользовательской метрикой расстояния между узлами графа. Кроме этого, в предложенной модели используется только одна матрицу эмбеддингов для узлов графа, в то время как в модели Skip-gram создаются отдельные матрицы для целей и контекстов.

Другая похожая модель — Global Vectors (GloVe) [94], которая моделирует вероятности совместной встречаемости в корпусе текстов. Целевая функция, которую

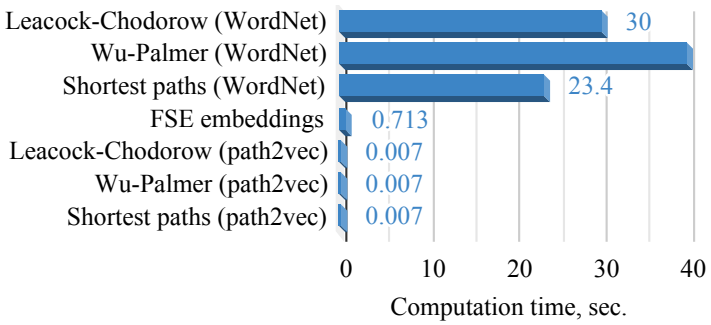

Рис. 9-1: Similarity computation: graph vs vectors.

необходимо минимизировать в модели GloVe, задается следующим образом:

$$\mathcal{L} = -\sum_{(v_i,v_j)\in B_p} \log\sigma(\mathbf{v}_i \cdot \tilde{\mathbf{v}}_j) - \sum_{(v_i,v_j)\in B_n} \log\sigma(-\mathbf{v}_i \cdot \tilde{\mathbf{v}}_j), \tag{9.2}$$

где s_{ij} – это количество совместных упоминаний слова v_i и v_j, b_i и b_j — это свободные параметры, а $f(s_{ij})$ — это весовая функция, учитывающая редкие события. Как и модель Skip-gram, GloVe также использует две матрицы эмбеддингов, но полагается только на глобальную информацию, предварительно агрегируя глобальную информацию о совместной встречаемости слов в корпусе.

Вычисление обучающей выборки

В общем случае, предложенная модель для обучения требует вычисления попарного сходства узлов s_{ij} между всеми парами узлов во входном графе G. Этот шаг может оказаться дорогостоящим в вычислительном отношении, однако он выполняется только один раз для ускорения вычисление сходства между узлами. Для некоторых метрик существуют эффективные алгоритмы, вычисляющие все попарные сходства одновременно, например [95] алгоритм вычисления расстояний по кратчайшим путям с производительностью в худшем случае $O(|V|^2 \log|V| + |V||E|)$. Поскольку входной набор обучающих примеров также растет квадратично по $|V|$, время обучения для больших графов может быть значительным. Чтобы решить эту проблему, возможно сократить входной обучающую выборку таким образом, чтобы каждый узел $v_i \in V$ имел только $k \in [50; 200]$ наиболее похожих узлов. Такое преобразование не приводит к потере эффективности, согласно нашим экспериментам.

9.3 Результаты

Были проведены эксперименты для измерения вычислительной эффективности алгоритма, а также качества аппроксимации на основе набора данных SimLex999 и на основе наборов данных SemEval по дизамбигуации. Было проведено сравнение с базовыми подходами, такими как Deepwalk, node2vec или TransR. Было продемонстрировано, что этот подход хорошо обобщается для графов (WordNet, Freebase и DBpedia). Кроме того, метод был интегрирован в графовый алгоритм дизамбигуации, показав, что его векторизованный аналог дает сопоставимые оценки F1 для этой задачи.

Path2vec обеспечивает ускорение вычисления расстояний на графике до четырех порядков по сравнению с оригинальными вычислениями на графе (см. Рис. 9-1). Модель можно применить к любому графу для которого задана мерой расстояния для ускорения алгоритмов, использующих расстояния в графе.

Детали экспериментальных результатов и их анализ можно найти в [3].

Глава 10

Лексические замены и анализ типов семантических отношений

Материалы данной главы основаны на статье [2] из списка 14 публикаций на которых основана диссертация.

10.1 Введение

В этой главе представлены методы лексической замены и их анализ.

Лексическая замена — это задача подбора слов, которые могут заменить выбранное слово в заданном текстовом контексте. Например, в предложении "*My daughter purchased a new car*" слово *car* можно заменить его синонимом *automobile*, а также когипонимом *bike*. или даже гипернимом *автомобиль*, сохраняя грамматику исходного предложения. Более формально, задача лексической замены формулируется следующим образом. Для заданного **предложения** S состоящего из **контекста** C и **целевого слова** T найти **лексические замены** (или лексические подстановки): слова или фразы, которые могут быть использованы вместо T без изменения смысла S как показано ниже:

- "We were not able to travel in the weather, and there was no phone." → telephone;
- "What happened to the big, new garbage can at Church and Chambers Streets?" → bin, disposal, container.

Генерация лексических подстановок является мощной технологией, которую можно использовать в качестве основы различных приложений обработки текстов, таких как индукция смысла слова [96], извлечение лексических отношений [97], генерация парафразов, упрощение текста, синтезирование текстовых данных и т.п. При этом, предпочтительный тип замены (например, замена на синонимы, гиперонимы, когипонимы) зависит от поставленной задачи.

В этом разделе представлено исследование методов лексической замены с использованием как классических, так и более современных языковых и маскированных языковых моделей (LM и MLM), таких как context2vec, ELMo, BERT, RoBERTa, XLNet. Показано, что уже конкурентоспособные результаты, достигнутые с помощью LM/MLM, могут быть дополнительно существенно улучшены, если информация о целевом слове предоставляется модели. Кроме того, проводится анализ типов лексико-семантических отношений, генерируемых этими моделями, как показано на Рис. 10-1.

We were not able to travel in the weather , and there was no ***phone*** .										
GOLD	**telephone (5)**									
OOC	phone	**telephone**	phones	cellphone	fone	videophone	handset	telephones	p990i	cell-phone
XLNet	electricity	internet	phone	power	**telephone**	car	water	communication	radio	tv
XLNet+embs	phone	**telephone**	phones	cellphone	internet	radio	electricity	iphone	car	computer
What happened to the big , new garbage ***can*** at Church and Chambers Streets ?										
GOLD	**bin (4)**	**disposal (1)**	**container (1)**							
OOC	can	could	should	would	will	must	might	to	may	ll
XLNet	can	dump	**bin**	truck	**disposal**	pit	heap	pile	**container**	stand
XLNet+embs	can	could	will	**bin**	cannot	dump	may	truck	**disposal**	stand

Типы семантических отношений: синоним, когипоним, когипоним 3, целевое слово, прямой гипероним, транзитивный гипероним, прямой гипоним, транзитивный гипоним, неизвестный тип, неизвестное слово

Рис. 10-1: Примеры лучших замен предоставленных аннотаторами (GOLD), базовым методом (OOC) и двумя разработанными моделями (XLNet и XLNet+embs). Целевое слово в каждом предложении выделено полужирным шрифтом, верные замены также выделены полужирным шрифтом. Каждая замена окрашен в соответствии с типом его семантического отношения к целевому слову.

10.2 Метод

Для генерации замен мы вводим несколько методов замен, которые представляют собой модели, принимающие фрагмент текста и позицию целевого слова в нем в качестве входных данных и генерируют список замен с их вероятностями. Для

построения наших методов мы используем следующие языковые модели (LM/MLM): context2vec [98], ELMo [99], BERT [100], RoBERTa [101] и XLNet [102].

Для заданного целевого слова, базовый подход для моделей, таких как context2vec и ELMo, заключается в кодировании его контекста и предсказании распределения вероятностей по возможным словам в заданном контексте. Таким образом, модель не использует информацию о целевом слове. Для маскированных языковых моделей (MLM) того же результата можно добиться, замаскировав целевое слово. Этот базовый подход использует основную способность LM/MLM предсказывать слова, соответствующие конкретному контексту. Однако эти слова часто не связаны с целью замены. Информация о целевом слове может улучшить сгенерированные замены.

Мы описываем метод введения информации об исходном целевом слове в нейронные модели лексической замены. Предположим, у нас есть пример LTR, где T — целевое слово, а $C = (L, R)$ — его контекст (левый и правый соответственно). Например, появление целевого слова *fly* в предложении "*Let me fly away!*" будет представлено как $T =$*"fly"*, $L =$*"'Let me"*, $R =$*"away!"*.

Предложенный метод комбинирует распределение вероятности замены при условии контекста $P(s|C)$ с распределением, основанным на близости возможных замен к целевому слов $P(s|T)$. Близость вычисляется как скалярное произведение между соответствующими вектоармм, а функция софтмакс применяется для получения распределения вероятностей. Чтобы выровнять порядки распределений, мы используем параметр температуры:

$$P(s|T) \propto \exp(\frac{\langle emb_s, emb_T \rangle}{\mathcal{T}}). \tag{10.1}$$

Методы инъекции целевых слов основаны на сходстве векторов слов и обозначаются как "+embs". Окончательное распределение получается по формуле

$$P(s|C,T) \propto \frac{P(s|C)P(s|T)}{P(s)^\beta}. \tag{10.2}$$

Для $\beta = 1$ данное выражение может быть получено с использованием формулы Байеса в предположении контекстной независимости C и T относительно s. Другие значения β могут быть использованы для пенализации частотных слов. Вероятности слов $P(s)$ оцениваются из частот слов для всех моделей кроме модели ELMo.

Согласно [103], для ELMo вероятности слов вычисляется на основе рангов слов в словаре ELMo (который упорядочен по частоте слов) на основе распределения Ципфа-Мандельброта.

Различные LM/MLM используются, как описано ниже, для получения контекстно-ориентированного распределения вероятностей подстановки $P(s|C)$. Для каждого из них мы экспериментируем с разными методами внедрения целей: context2vec, ELMo, BERT/RoBERTa, XLNet.

В дополнение к этому методу были протестированы более простые стратегии инъекции информации о целевом слова, такие как динамические шаблоны, дублирование целевого слова и другие. Однако эти методы показали более слабые результаты, чем метод описанный выше.

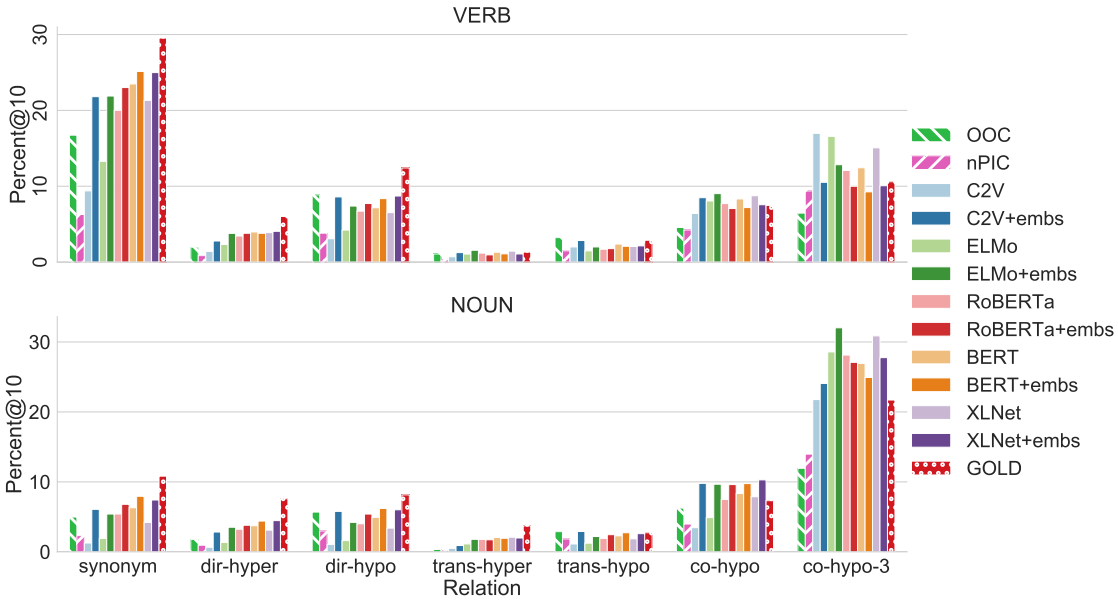

Рис. 10-2: Статистика типов лексических замен, связанных с целью замены различными семантическими отношениями по данным WordNet.

10.3 Результаты

Предложенные методы были протестированы на датасетах лексических замен (SemEval-2007 и CoInCo) и на задаче извлечения смысл слов (SemEval-2010 и SemEval-2013). Экспериментальные результаты на двух наборах данных показывают,

что предлагаемый метод лексической замены с инъекцией информации о целевом слове (+embs) неизменно превосходит базовые методы и модели, такие как C2V, ELMo, BERT, RoBERTa, XLNet.

Кроме того, в был представлен анализ типов семантических отношений, таких как синонимия, гипернимия и когипонимия, между целевыми словами и их заменами, созданными с помощью различных моделей, как показано на Рис. 10-2. Результаты показывают, что среди замен существительных доминируют когипонимы, тогда как для глаголов доминируют синонимы, причем когипонимы представляют второй по величине тип.

Детали экспериментальных результатов и их анализ можно найти в [2].

Глава 11

Заключение

По результатам исследований, вошедших в данную диссертацию, были опубликованы статьи по методам и алгоритмам вычислительной лексической семантики для извлечения смыслов слов, гипернимических отношений между словами и семантических фреймов [1–42]. В частности, положения, защищенные в этой диссертации, основаны на 14 публикациях [1–10, 17–20].

Набор полученных результатов создал основу вычислительной обработки значений слов и отношений между ними. Перечислим основные результаты, полученные в данной диссертации, которые выносятся на защиту:

- **Алгоритм кластеризации графов:** Нечеткий мета-алгоритм кластеризации для обработки больших лингвистических графов. Алгоритм применен для извлечения синсетов, семантических фреймов и классов.

- **Векторные представления смыслов слов:** Методы получения векторных представления смыслов слов с использованием кластеризации графов. Метод дизамбигуации с помощью этих представлений.

- **Создание интерпретируемых представлений значений слов:** Методы создания интерпретируемых представлений значений слов путем автоматического поиска гипернимов, изображений и определений.

- **Выравнивание смысловых представлений слов:** Метод выравнивания созданных вручную и автоматически индуцированных смыслов слов.

- **Дизамбигуация слов в контексте:** Методы устранения неоднозначности значения слова в контексте и методы лексической замены в контексте.

- **Построение семантических деревьев:** Метод для улучшения построенных вручную таксономий за счет добавления новых и удаления неверных отношений гиперонимии. Метод очистки автоматически извлеченных отношений гипернимии, основанный на кластеризации графов смыслов слов.

- **Векторизация лексико-семантических графов:** Методы векторизации узлов графов за счет приближения графовых метрик.

Большинство разработанных методов полагаются на представления графов, поскольку лексико-семантические ресурсы естественным образом представляются в виде графов, где узлы представляют собой смыслы слов, а семантические отношения являются ребрами. В то же время векторные представления также широко используются, демонстрируя двойственность графово-векторного представления для различных задач вычислительной лексической семантики. С одной стороны, разработанные методы могут быть использованы для автоматизации работы ручного труда лексикографов, создающих и поддерживающих в актуальном состоянии различные лексико-семантические ресурсы. С другой стороны, их можно использовать для улучшения интерпретируемости нейронных моделей путем связывания векторных представлений с интерпретируемыми представлениями на основе графов. Кроме того, такое связывание может повысить качество приложений за счет извлечения дополнительных признаков из графовых представлений.

Перспективным направлением будущей работы является исследование языковых моделей, таких как T5, GPT или LLaMa, для задачи генерации и пополнения лексико-семантических ресурсов и других задач, связанных с моделированием значения значений слов и отношений между ними. Некоторые результаты работы в этом направлении уже были опубликованы автором в работах [40, 104–106], что подтверждает перспективность дальшейших исследований и разработок с задач, описанных в диссертации с использованием больших языковых моделей.

Литература

[1] A. Panchenko, E. Ruppert, S. Faralli, S. P. Ponzetto, and C. Biemann, "**Unsupervised does not mean uninterpretable: The case for word sense induction and disambiguation**," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, (Valencia, Spain), pp. 86–98, Association for Computational Linguistics, Apr. 2017.

[2] N. Arefyev, B. Sheludko, A. Podolskiy, and A. Panchenko, "**Always Keep your Target in Mind: Studying Semantics and Improving Performance of Neural Lexical Substitution**," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 1242–1255, International Committee on Computational Linguistics, Dec. 2020.

[3] A. Kutuzov, M. Dorgham, O. Oliynyk, C. Biemann, and A. Panchenko, "**Making Fast Graph-based Algorithms with Graph Metric Embeddings**," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 3349–3355, Association for Computational Linguistics, July 2019.

[4] D. Ustalov, N. Arefyev, C. Biemann, and A. Panchenko, "**Negative Sampling Improves Hypernymy Extraction Based on Projection Learning**," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, (Valencia, Spain), pp. 543–550, Association for Computational Linguistics, Apr. 2017.

[5] R. Aly, S. Acharya, A. Ossa, A. Köhn, C. Biemann, and A. Panchenko, "**Every Child Should Have Parents: A Taxonomy Refinement Algorithm Based on Hyperbolic Term Embeddings**," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 4811–4817, Association for Computational Linguistics, July 2019.

[6] D. Ustalov, A. Panchenko, C. Biemann, and S. P. Ponzetto, "**Watset: Local-Global Graph Clustering with Applications in Sense and Frame Induction**," *Computational Linguistics*, vol. 45, pp. 423–479, Sept. 2019.

[7] S. Faralli, A. Panchenko, C. Biemann, and S. P. Ponzetto, "**Linked Disambiguated Distributional Semantic Networks**," in *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II* (P. Groth, E. Simperl, A. J. G. Gray, M. Sabou, M. Krötzsch, F. Lécué, F. Flöck, and Y. Gil, eds.), vol. 9982 of *Lecture Notes in Computer Science*, pp. 56–64, 2016.

[8] C. Biemann, S. Faralli, A. Panchenko, and S. P. Ponzetto, "**A framework for enriching lexical semantic resources with distributional semantics**," *Nat. Lang. Eng.*, vol. 24, no. 2, pp. 265–312, 2018.

[9] D. Ustalov, A. Panchenko, and C. Biemann, "**Watset: Automatic Induction of Synsets from a Graph of Synonyms**," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Vancouver, Canada), pp. 1579–1590, Association for Computational Linguistics, July 2017.

[10] D. Ustalov, A. Panchenko, A. Kutuzov, C. Biemann, and S. P. Ponzetto, "**Unsupervised Semantic Frame Induction using Triclustering**," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Melbourne, Australia), pp. 55–62, Association for Computational Linguistics, July 2018.

[11] Ö. Sevgili, A. Shelmanov, M. Y. Arkhipov, A. Panchenko, and C. Biemann, "**Neural entity linking: A survey of models based on deep learning**," *Semantic Web*, vol. 13, no. 3, pp. 527–570, 2022.

[12] S. Anwar, A. Shelmanov, N. Arefyev, A. Panchenko, and C. Biemann, "**Text augmentation for semantic frame induction and parsing**," *Language Resources and Evaluation*, vol. 23, no. 3, pp. 527–556, 2023.

[13] A. Jana, D. Puzyrev, A. Panchenko, P. Goyal, C. Biemann, and A. Mukherjee, "**On the Compositionality Prediction of Noun Phrases using Poincaré Embeddings**," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 3263–3274, Association for Computational Linguistics, July 2019.

[14] I. Nikishina, V. Logacheva, A. Panchenko, and N. Loukachevitch, "**Studying Taxonomy Enrichment on Diachronic WordNet Versions**," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 3095–3106, International Committee on Computational Linguistics, Dec. 2020.

[15] I. Nikishina, M. Tikhomirov, V. Logacheva, Y. Nazarov, A. Panchenko, and N. V. Loukachevitch, "**Taxonomy enrichment with text and graph vector representations**," *Semantic Web*, vol. 13, no. 3, pp. 441–475, 2022.

[16] S. Faralli, A. Panchenko, C. Biemann, and S. P. Ponzetto, "**The ContrastMedium Algorithm: Taxonomy Induction From Noisy Knowledge Graphs With Just A Few Links**," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, (Valencia, Spain), pp. 590–600, Association for Computational Linguistics, Apr. 2017.

[17] M. Pelevina, N. Arefiev, C. Biemann, and A. Panchenko, "**Making Sense of Word Embeddings**," in *Proceedings of the 1st Workshop on Representation Learning for NLP*, (Berlin, Germany), pp. 174–183, Association for Computational Linguistics, Aug. 2016.

[18] A. Panchenko, F. Marten, E. Ruppert, S. Faralli, D. Ustalov, S. P. Ponzetto, and C. Biemann, "**Unsupervised, Knowledge-Free, and Interpretable Word Sense Disambiguation**," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, (Copenhagen, Denmark), pp. 91–96, Association for Computational Linguistics, Sept. 2017.

[19] A. Panchenko, D. Ustalov, S. Faralli, S. P. Ponzetto, and C. Biemann, "**Improving Hypernymy Extraction with Distributional Semantic Classes**," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018.

[20] V. Logacheva, D. Teslenko, A. Shelmanov, S. Remus, D. Ustalov, A. Kutuzov, E. Artemova, C. Biemann, S. P. Ponzetto, and A. Panchenko, "**Word Sense Disambiguation for 158 Languages using Word Embeddings Only**," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, (Marseille, France), pp. 5943–5952, European Language Resources Association, May 2020.

[21] A. Panchenko, S. Faralli, E. Ruppert, S. Remus, H. Naets, C. Fairon, S. P. Ponzetto, and C. Biemann, "**TAXI at SemEval-2016 Task 13: a Taxonomy Induction Method based on Lexico-Syntactic Patterns, Substrings and Focused Crawling**," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, (San Diego, California), pp. 1320–1327, Association for Computational Linguistics, June 2016.

[22] S. Anwar, A. Shelmanov, A. Panchenko, and C. Biemann, "**Generating Lexical Representations of Frames using Lexical Substitution**," in *Proceedings of the Probability and Meaning Conference (PaM 2020)*, (Gothenburg), pp. 95–103, Association for Computational Linguistics, June 2020.

[23] D. Ustalov, A. Panchenko, C. Biemann, and S. P. Ponzetto, "**Unsupervised Sense-Aware Hypernymy Extraction**," in *Proceedings of the 14th Conference on Natural Language Processing, KONVENS 2018, Vienna, Austria, September 19-21, 2018* (A. Barbaresi, H. Biber, F. Neubarth, and R. Osswald, eds.), pp. 192–201, Österreichische Akademie der Wissenschaften, 2018.

[24] A. Panchenko, A. Lopukhina, D. Ustalov, K. Lopukhin, N. Arefyev, A. Leontyev, and N. V. Loukachevitch, "**RUSSE'2018: A Shared Task on Word Sense Induction for the Russian Language**," in *Proceedings of the 24th International Conference on Computational Linguistics and Intellectual Technologies (Dialogue'2018). Moscow, Russia.*, pp. 192–201, RGGU, 2018.

[25] N. Arefyev, P. Ermolaev, and A. Panchenko, "**How much does a word weigh? Weighting word embeddings for word sense induction**," in *Proceedings of the 24th International Conference on Computational Linguistics and Intellectual Technologies (Dialogue'2018). Moscow, Russia.*, pp. 201–212, RGGU, 2018.

[26] D. Ustalov, D. Teslenko, A. Panchenko, M. Chernoskutov, C. Biemann, and S. P. Ponzetto, "**An Unsupervised Word Sense Disambiguation System for Under-Resourced Languages**," in *Proceedings of the Eleventh International*

Conference on Language Resources and Evaluation (LREC 2018), (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018.

[27] S. Faralli, A. Panchenko, C. Biemann, and S. P. Ponzetto, "**Enriching Frame Representations with Distributionally Induced Senses**," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018.

[28] Ö. Sevgili, A. Panchenko, and C. Biemann, "**Improving Neural Entity Disambiguation with Graph Embeddings**," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, (Florence, Italy), pp. 315–322, Association for Computational Linguistics, July 2019.

[29] A. Panchenko, "**Best of Both Worlds: Making Word Sense Embeddings Interpretable**," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, (Portorož, Slovenia), pp. 2649–2655, European Language Resources Association (ELRA), May 2016.

[30] D. Ustalov, M. Chernoskutov, C. Biemann, and A. Panchenko, "**Fighting with the Sparsity of Synonymy Dictionaries for Automatic Synset Induction**," in *Analysis of Images, Social Networks and Texts - 6th International Conference, AIST 2017, Moscow, Russia, July 27-29, 2017, Revised Selected Papers* (W. M. P. van der Aalst, D. I. Ignatov, M. Y. Khachay, S. O. Kuznetsov, V. S. Lempitsky, I. A. Lomazova, N. V. Loukachevitch, A. Napoli, A. Panchenko, P. M. Pardalos, A. V. Savchenko, and S. Wasserman, eds.), vol. 10716 of *Lecture Notes in Computer Science*, pp. 94–105, Springer, 2017.

[31] A. Panchenko, J. Simon, M. Riedl, and C. Biemann, "**Noun Sense Induction and Disambiguation using Graph-Based Distributional Semantics**," in *Proceedings of the 13th Conference on Natural Language Processing, KONVENS 2016, Bochum, Germany, September 19-21, 2016* (S. Dipper, F. Neubarth, and H. Zinsmeister, eds.), vol. 16 of *Bochumer Linguistische Arbeitsberichte*, 2016.

[32] D. Puzyrev, A. Shelmanov, A. Panchenko, and E. Artemova, "**Noun Compositionality Detection Using Distributional Semantics for the Russian Language**," in *Analysis of Images, Social Networks and Texts - 8th International Conference, AIST 2019, Kazan, Russia, July 17-19, 2019, Revised Selected Papers* (W. M. P. van der Aalst, V. Batagelj, D. I. Ignatov, M. Y. Khachay, V. V. Kuskova, A. Kutuzov, S. O. Kuznetsov, I. A. Lomazova, N. V. Loukachevitch, A. Napoli, P. M. Pardalos, M. Pelillo, A. V. Savchenko, and E. Tutubalina, eds.), vol. 11832 of *Lecture Notes in Computer Science*, pp. 218–229, Springer, 2019.

[33] N. Arefyev, B. Sheludko, and A. Panchenko, "**Combining Lexical Substitutes in Neural Word Sense Induction**," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, (Varna, Bulgaria), pp. 62–70, INCOMA Ltd., Sept. 2019.

[34] A. Kutuzov, M. Dorgham, O. Oliynyk, C. Biemann, and A. Panchenko, "**Learning Graph Embeddings from WordNet-based Similarity Measures**," in

*Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, (Minneapolis, Minnesota), pp. 125–135, Association for Computational Linguistics, June 2019.

[35] A. Razzhigaev, N. Arefyev, and A. Panchenko, "**SkoltechNLP at SemEval-2021 Task 2: Generating Cross-Lingual Training Data for the Word-in-Context Task**," in *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, (Online), pp. 157–162, Association for Computational Linguistics, Aug. 2021.

[36] D. Puzyrev, A. Shelmanov, A. Panchenko, and E. Artemova, "**A Dataset for Noun Compositionality Detection for a Slavic Language**," in *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, (Florence, Italy), pp. 56–62, Association for Computational Linguistics, Aug. 2019.

[37] N. Arefyev, B. Sheludko, A. Davletov, D. Kharchev, A. Nevidomsky, and A. Panchenko, "**Neural GRANNy at SemEval-2019 Task 2: A combined approach for better modeling of semantic relationships in semantic frame induction**," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, (Minneapolis, Minnesota, USA), pp. 31–38, Association for Computational Linguistics, June 2019.

[38] S. Anwar, D. Ustalov, N. Arefyev, S. P. Ponzetto, C. Biemann, and A. Panchenko, "**HHMM at SemEval-2019 Task 2: Unsupervised Frame Induction using Contextualized Word Embeddings**," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, (Minneapolis, Minnesota, USA), pp. 125–129, Association for Computational Linguistics, June 2019.

[39] A. Panchenko, S. Faralli, S. P. Ponzetto, and C. Biemann, "**Using Linked Disambiguated Distributional Networks for Word Sense Disambiguation**," in *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, (Valencia, Spain), pp. 72–78, Association for Computational Linguistics, Apr. 2017.

[40] I. Nikishina, I. Andrianov, A. Vakhitova, and A. Panchenko, "TaxFree: a visualization tool for candidate-free taxonomy enrichment," in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations* (W. Buntine and M. Liakata, eds.), (Taipei, Taiwan), pp. 39–47, Association for Computational Linguistics, Nov. 2022.

[41] I. Nikishina, N. Loukachevitch, V. Logacheva, and A. Panchenko, "**Evaluation of Taxonomy Enrichment on Diachronic WordNet Versions**," in *Proceedings of the 11th Global Wordnet Conference*, (University of South Africa (UNISA)), pp. 126–136, Global Wordnet Association, Jan. 2021.

[42] I. Nikishina, A. Vakhitova, E. Tutubalina, and A. Panchenko, "**Cross-Modal Contextualized Hidden State Projection Method for Expanding of Taxonomic Graphs**," in *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, (Gyeongju, Republic of Korea), pp. 11–24, Association for Computational Linguistics, Oct. 2022.

[43] B. Dorow and D. Widdows, "Discovering Corpus-Specific Word Senses," in *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 2*, EACL '03, (Budapest, Hungary), pp. 79–82, Association for Computational Linguistics, 2003.

[44] J. Véronis, "HyperLex: lexical cartography for information retrieval," *Computer Speech & Language*, vol. 18, no. 3, pp. 223–252, 2004.

[45] C. Biemann, "Chinese Whispers: An Efficient Graph Clustering Algorithm and Its Application to Natural Language Processing Problems," in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, (New York, NY, USA), pp. 73–80, Association for Computational Linguistics, 2006.

[46] D. Hope and B. Keller, "MaxMax: A Graph-Based Soft Clustering Algorithm Applied to Word Sense Induction," in *Computational Linguistics and Intelligent Text Processing: 14th International Conference, CICLing 2013, Samos, Greece, March 24-30, 2013, Proceedings, Part I*, vol. 7816 of *Lecture Notes in Computer Science*, pp. 368–381, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[47] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Workshop at International Conference on Learning Representations (ICLR)*, (Scottsdale, AZ, USA), pp. 1310–1318, 2013.

[48] A. Joulin, E. Grave, P. Bojanowski, M. Nickel, and T. Mikolov, "Fast linear model for knowledge graph embeddings," *arXiv preprint arXiv:1710.10881*, 2017.

[49] C. Biemann and M. Riedl, "Text: Now in 2D! a framework for lexical expansion with contextual similarity," *Journal of Language Modelling*, vol. 1, no. 1, pp. 55–95, 2013.

[50] C. Biemann, "Chinese Whispers: an efficient graph clustering algorithm and its application to natural language processing problems," in *Proceedings of the first workshop on graph based methods for natural language processing*, TextGraphs-1, (New York City, NY, USA), pp. 73–80, Association for Computational Linguistics, 2006.

[51] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems 26*, NIPS 2013, pp. 3111–3119, Harrahs and Harveys, NV, USA: Curran Associates, Inc., 2013.

[52] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning Word Vectors for 157 Languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, (Miyazaki, Japan), pp. 3483–3487, European Language Resources Association (ELRA), 2018.

[53] A. Vellido, J. D. Martín, F. Rossi, and P. J. Lisboa, "Seeing is believing: The importance of visualization in real-world machine learning applications," in *Proceedings of the 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2011)*, (Bruges, Belgium), pp. 219–226, 2011.

[54] A. A. Freitas, "Comprehensible classification models: a position paper," *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 1, pp. 1–10, 2014.

[55] J. Li, X. Chen, E. Hovy, and D. Jurafsky, "Visualizing and Understanding Neural Models in NLP," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (San Diego, CA, USA), pp. 681–691, Association for Computational Linguistics, June 2016.

[56] C. Biemann, "Turk Bootstrap Word Sense Inventory 2.0: A Large-Scale Resource for Lexical Substitution," in *Proceedings of the 8th International Conference on Language Resources and Evaluation*, (Istanbul, Turkey), pp. 4038–4042, European Language Resources Association, 2012.

[57] D. Jurgens and I. Klapaftis, "Semeval-2013 Task 13: Word Sense Induction for Graded and Non-graded Senses," in *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval'2013)*, vol. 2, (Montreal, Canada), pp. 290–299, Association for Computational Linguistics, 2013.

[58] S. P. Ponzetto and R. Navigli, "Knowledge-rich word sense disambiguation rivaling supervised systems," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL'10, (Uppsala, Sweden), pp. 1522–1531, Association for Computational Linguistics, 2010.

[59] I. Gurevych, J. Eckle-Kohler, S. Hartmann, M. Matuschek, C. M. Meyer, and C. Wirth, "UBY - A Large-Scale Unified Lexical-Semantic Resource Based on LMF," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL'12, (Avignon, France), pp. 580–590, Association for Computational Linguistics, 2012.

[60] S. Pavel and J. Euzenat, "Ontology Matching: State of the Art and Future Challenges," *IEEE Transaction on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 158–176, 2013.

[61] R. Navigli and S. P. Ponzetto, "BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network," *Artificial Intelligence*, vol. 193, pp. 217–250, 2012.

[62] R. Fu, J. Guo, B. Qin, W. Che, H. Wang, and T. Liu, "Learning Semantic Hierarchies via Word Embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Baltimore, MD, USA), pp. 1199–1209, Association for Computational Linguistics, 2014.

[63] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, (Berkeley, California, USA), pp. 281–297, University of California Press, 1967.

[64] T. Wandmacher, "How semantic is Latent Semantic Analysis?," in *Proceedings of RÉCITAL 2005*, (Dourdan, France), pp. 525–534, 2005.

[65] K. Heylen, Y. Peirsman, D. Geeraerts, and D. Speelman, "Modelling Word Similarity: an Evaluation of Automatic Synonymy Extraction Algorithms," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*,

LREC 2008, (Marrakech, Morocco), pp. 3243–3249, European Language Resources Association (ELRA), 2008.

[66] A. Panchenko, "Comparison of the Baseline Knowledge-, Corpus-, and Web-based Similarity Measures for Semantic Relations Extraction," in *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, (Edinburgh, UK), pp. 11–21, Association for Computational Linguistics, 2011.

[67] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, vol. abs/1412.6980, 2014.

[68] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *CoRR*, vol. abs/1603.04467, 2016.

[69] S. Faralli, A. Panchenko, C. Biemann, and S. P. Ponzetto, "Linked Disambiguated Distributional Semantic Networks," in *The Semantic Web – ISWC 2016: 15th International Semantic Web Conference, Proceedings, Part II*, Lecture Notes in Computer Science, pp. 56–64, Kobe, Japan: Springer International Publishing, 2016.

[70] C. Biemann, S. Faralli, A. Panchenko, and S. P. Ponzetto, "A framework for enriching lexical semantic resources with distributional semantics," *Natural Language Engineering*, pp. 1–48, 2018.

[71] C. Biemann and M. Riedl, "Text: now in 2D! A framework for lexical expansion with contextual similarity," *Journal of Language Modelling*, vol. 1, no. 1, pp. 55–95, 2013.

[72] D. Widdows and B. Dorow, "A graph model for unsupervised lexical acquisition," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pp. 1–7, Association for Computational Linguistics, 2002.

[73] M. Everett and S. P. Borgatti, "Ego network betweenness," *Social networks*, vol. 27, no. 1, pp. 31–38, 2005.

[74] C. Fellbaum, *WordNet: An Electronic Database*. Cambridge, MA: MIT Press, 1998.

[75] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artificial Intelligence*, vol. 193, pp. 217–250, 2012.

[76] M. Nickel and D. Kiela, "Poincaré Embeddings for Learning Hierarchical Representations," in *Advances in Neural Information Processing Systems 30*, (Long Beach, CA, USA), pp. 6338–6347, 2017.

[77] R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda, "English gigaword forth edition," in *Linguistic Data Consortium*, (Philadelphia, PA, USA), 2009.

[78] A. Ferraresi, E. Zanchetta, M. Baroni, and S. Bernardini, "Introducing and evaluating ukWaC, a very large web-derived corpus of English," in *Proceedings of the 4th Web as Corpus Workshop. Can we beat Google?*, (Marrakech, Morocco), pp. 47–54, 2008.

[79] D. Goldhahn, T. Eckart, and U. Quasthoff, "Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages," in *Proceedings of the Eight International Conference on Language Resources and Evaluation*, (Istanbul, Turkey), pp. 759–765, 2012.

[80] A. Panchenko, O. Morozova, and H. Naets, "A semantic similarity measure based on lexico-syntactic patterns," in *Proceedings of KONVENS 2012*, (Vienna, Austria), pp. 174–178, September 2012.

[81] J. Seitner, C. Bizer, K. Eckert, S. Faralli, R. M. und Heiko Paulheim, and S. P. Ponzetto, "A Large DataBase of Hypernymy Relations Extracted from the Web," in *Proceedings of the 10th International Conference on Language Resources and Evaluation*, (Portorož, Slovenia), pp. 360–367, 2016.

[82] A. Panchenko, S. Faralli, E. Ruppert, S. Remus, H. Naets, C. Fairon, S. P. Ponzetto, and C. Biemann, "TAXI at SemEval-2016 Task 13: a taxonomy Induction Method based on Lexico-syntactic Patterns, Substrings and Focused Crawling," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, SemEval@NAACL-HLT'16, (San Diego, CA, USA), pp. 1320–1327, Association for Computational Linguistics, 2016.

[83] J. Stillwell, *Sources of hyperbolic geometry*. No. History of Mathematics, Volume 10 in 1, American Mathematical Society, 1996.

[84] K. Heylen, Y. Peirsman, D. Geeraerts, and D. Speelman, "Modelling word similarity: an evaluation of automatic synonymy extraction algorithms," in *Proceedings of the sixth international language resources and evaluation*, (Marrakech, Morocco), pp. 3243–3249, 2008.

[85] J. Weeds, D. Clarke, J. Reffin, D. Weir, and B. Keller, "Learning to distinguish hypernyms and co-hyponyms," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*, (Dublin, Ireland), pp. 2249–2259, Dublin City University and Association for Computational Linguistics, 2014.

[86] G. Bordea, P. Buitelaar, S. Faralli, and R. Navigli, "Semeval-2015 task 17: Taxonomy Extraction Evaluation (TExEval)," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, (Denver, CO, USA), pp. 902–910, 2015.

[87] R. Tarjan, "Depth first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146 –160, 1972.

[88] G. A. Miller, "WordNet: A Lexical Database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[89] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on knowledge and data engineering*, vol. 19, no. 3, pp. 355–369, 2007.

[90] M. T. Pilehvar and R. Navigli, "From senses to texts: An all-in-one graph-based approach for measuring semantic similarity," *Artificial Intelligence*, vol. 228, pp. 95–128, 2015.

[91] B. Lebichot, G. Guex, I. Kivimäki, and M. Saerens, "A Constrained Randomized Shortest-Paths Framework for Optimal Exploration," *arXiv preprint arXiv:1807.04551*, 2018.

[92] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, (San Diego, CA, USA), 2015.

[93] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems 26*, (Lake Tahoe, NV, USA), pp. 3111–3119, Curran Associates, Inc., 2013.

[94] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, 2014.

[95] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM (JACM)*, vol. 24, no. 1, pp. 1–13, 1977.

[96] A. Amrami and Y. 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), pp. 4860–4867, Association for Computational Linguistics, Oct.-Nov. 2018.

[97] T. Schick and H. Schütze, "Rare Words: A Major Problem for Contextualized Embeddings and How to Fix it by Attentive Mimicking.," in *AAAI*, pp. 8766–8774, 2020.

[98] O. Melamud, J. Goldberger, and I. Dagan, "context2vec: Learning Generic Context Embedding with Bidirectional LSTM," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, (Berlin, Germany), pp. 51–61, Association for Computational Linguistics, Aug. 2016.

[99] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep Contextualized Word Representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.

[100] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[101] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *ArXiv*, vol. abs/1907.11692, 2019.

[102] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in neural information processing systems*, pp. 5753–5763, 2019.

[103] N. Arefyev, B. Sheludko, and A. Panchenko, "Combining Lexical Substitutes in Neural Word Sense Induction," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'19)*, RANLP '19, (Varna, Bulgaria), pp. 62–70, 2019.

[104] V. Moskvoretskii, E. Neminova, A. Lobanova, A. Panchenko, and I. Nikishina, "TaxoLLaMA: WordNet-based model for solving multiple lexical semantic tasks," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (L.-W. Ku, A. Martins, and V. Srikumar, eds.), (Bangkok, Thailand), pp. 2331–2350, Association for Computational Linguistics, Aug. 2024.

[105] V. Moskvoretskii, A. Panchenko, and I. Nikishina, "Are large language models good at lexical semantics? a case of taxonomy learning," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, eds.), (Torino, Italia), pp. 1498–1510, ELRA and ICCL, May 2024.

[106] P. Chernomorchenko, A. Panchenko, and I. Nikishina, "Leveraging taxonomic information from large language models for hyponymy prediction," in *Analysis of Images, Social Networks and Texts* (D. I. Ignatov, M. Khachay, A. Kutuzov, H. Madoyan, I. Makarov, I. Nikishina, A. Panchenko, M. Panov, P. M. Pardalos, A. V. Savchenko, E. Tsymbalov, E. Tutubalina, and S. Zagoruyko, eds.), (Cham), pp. 49–63, Springer Nature Switzerland, 2024.