

DOI: 10.33184/dokbsu-2026.3.16

Эмбединги в системах машинного перевода

Р. Г. Мифтахова*, А. Т. Ханов

Уфимский университет науки и технологий

Россия, Республика Башкортостан, 450076 г. Уфа, улица Заки Валиди, 32.

Email: miftahovar@yandex.ru

Рассматриваются основные подходы к построению эмбедингов – векторных представлений слов, позволяющих моделям машинного обучения учитывать семантические и синтаксические свойства языка. Представлен обзор ключевых методов, преимуществ и ограничений. Выполняется опыт с искусственным языком, состоящим из небольшого словаря, для выявления различий между one-hot кодированием и современными методами построения эмбедингов.

Ключевые слова: машинный перевод, нейронные сети, эмбединги, векторное представление слов.

Современные методы обработки естественного языка (NLP) основаны на представлении слов в виде числовых векторов – эмбедингов. Этот подход дает возможность моделям распознавать смысловую близость, взаимосвязи между словами и их окружением. Важно отметить, что эволюция к текущим методам создания эмбедингов представляла собой поэтапный процесс, отмеченный существенными концептуальными сдвигами.

Целью настоящего исследования является краткий обзор ключевых методологий разработки эмбедингов с демонстрацией их характеристик на наглядном, легко интерпретируемом примере. Первым и самым простым способом числового представления слов можно считать one-hot encoding. Его логика предельно проста: каждому слову в словаре назначается отдельная позиция в векторе, к примеру, словарь состоит из пяти слов – каждое слово получает вектор длиной пять, где одна координата равна единице, а остальные равны нулю. Например, если слово «мужчина» стоит первым в словаре, оно может быть представлено как [1, 0, 0, 0, 0], а слово «женщина» – как [0, 1, 0, 0, 0]. Такой подход удобен тем, что он однозначен и не требует обучения модели. Модель легко различает слова, поскольку каждое имеет свой уникальный «адрес» в словаре. Поэтому one-hot encoding можно считать базовым методом кодирования языковых единиц. По этой причине one-hot encoding можно рассматривать как технически простую форму кодирования языковых единиц. Однако его недостаток проявляется сразу, как только возникает задача анализа смысла. В one-hot encoding все разные слова оказываются одинаково различными. Например, слова «король»

и «королева» не будут считаться более близкими, чем «король» и «дом», хотя для человека эта связь очевидна. В этом смысле one-hot encoding демонстрирует весьма важную грань между формально-синтаксическим и семантическим представлением языка. Он отвечает на вопрос: «какое слово перед нами?», но практически не отвечает на вопрос: «на что похоже это слово?». Первыми такую ограниченность векторные модели информационного поиска пытались преодолеть, представляя вопросы и ответы как точки в пространстве признаков. В работе Салтона, Вонга и Янга целостные тексты рассматривались как векторы, которые можно сравнивать по степени близости [1, с. 613–615]. Это был важный шаг к геометрическому пониманию текста, хотя на уровне отдельных слов такая модель еще не давала полноценного представления о значении.

Следующий этап связан с count-based методами, т.е. подходами, основанными на подсчете частот. Их главная идея восходит к распределительному пониманию языка: значение слова можно частично определить по его окружению. Еще Зеллиг Харрис писал, что распределение языкового элемента следует понимать как совокупность всех его окружений [2, с. 146–147]. В более поздних моделях эта мысль получила математическое представление: если два слова часто встречаются рядом с похожими словами, значит, они могут быть близки по смыслу. В count-based подходах обычно строится матрица совместной встречаемости. По строкам в ней располагаются слова, а по столбцам – контексты или другие слова. На пересечении строки и столбца записывается число: сколько раз слово встретилось рядом с данным контекстом. Если, к примеру, слово «врач» часто появляется в окружении таких слов, как «пациент», «больница», «лечение», то его вектор будет отражать именно такую среду. И, соответственно, слово «медсестра», имеющее подобный набор соседей, также окажется близким слову «врач» в векторном пространстве. Подобные модели важны тем, что они уже переходят от простой идентификации слова к его статистическому описанию. В обзоре Терни и Пантела векторные модели семантики делятся по типу матриц, включая матрицы «слово – контекст» и «термин – документ» [3, с. 141–145]. Очевидно, что семантика здесь понимается не как заранее заданный словарь значений, а как структура частотных отношений в корпусе. Ограничение count-based методов состоит в том, что такие матрицы часто становятся очень большими и разреженными: большинство слов редко встречается рядом с большинством других слов. Кроме того, простые частоты не всегда хорошо отражают смысл. Часто встречающиеся служебные слова могут создавать много «шума», а редкие, но важные слова могут быть представлены недостаточно надежно. Поэтому на практике такие матрицы часто преобразуют: нормируют, взвешивают, уменьшают размерность. Тем не менее именно count-based методы подготовили почву для более поздней стадии развития эмбедингов: показали, что смысл можно искать не в слове отдельно, а в системе его употреблений.

Крупный прорыв произошел с появлением Word2Vec, связанного прежде всего с работами Томаша Миколова. В отличие от count-based методов, Word2Vec не строит заранее огромную таблицу встречаемости всех слов в обычном представлении. Вместо этого, он обучает компактные плотные векторы слов через предсказательную задачу. В модели Skip-gram слово используется для предсказания его контекста, т.е. соседних слов [4, с. 3112–3113]. Если слово много раз встречается в похожих окружениях, модель постепенно выстраивает его вектор так, чтобы он отражал эту частотность.

Главное отличие Word2Vec от one-hot encoding состоит в том, что слово получает не разреженный вектор с одной единицей, а плотный набор чисел. Эти координаты обычно не имеют простых названий вроде «пол», «статус» или «предметность». Однако в совокупности они задают положение слова в смысловом пространстве. Близость векторов начинает соответствовать близости употребления, а часто и близости значения.

Модель GloVe – Global Vectors for Word Representation – была предложена Джеффри Пеннингтоном, Ричардом Сочером и Кристофером Мэннингом. Ее можно рассматривать как попытку соединить два направления: глобальную статистику совместной встречаемости и идею плотных векторов. Авторы прямо указывают, что GloVe объединяет преимущества глобальной матричной факторизации и локальных контекстных методов [5, с. 1532–1534].

Если Word2Vec учится предсказывать ближайшие слова в контекстном окне, то GloVe делает акцент на общей картине совместной встречаемости во всем корпусе. В основе модели лежит матрица, где фиксируется, как часто слова появляются рядом друг с другом. Однако GloVe не просто хранит эту матрицу. Он обучает такие векторы, чтобы отношения между словами отражали отношения между их частотами совместной встречаемости [5, с. 1535–1537].

Например, слова «лед» и «пар» могут по-разному соотноситься со словами «твердый», «газ», «вода» и «температура». Важной становится не только сама частота, но и отношение частот: какие контексты характерны для одного слова и нехарактерны для другого. Благодаря этому GloVe хорошо показывает себя в задачах семантической близости и аналогий. Авторы подчеркивали, что полученное пространство содержит осмысленную внутреннюю структуру [6, с. 1537–1539].

GloVe сохраняет сильные стороны count-based подходов, потому что опирается на глобальную статистику корпуса, но при этом создает компактные плотные векторы, похожие по удобству использования на Word2Vec. Его ограничение похоже на ограничение большинства неконтекстных эмбеддингов: одно слово обычно имеет одно представление. Поэтому многозначность и зависимость смысла от конкретного предложения учитываются лишь частично. Тем не менее GloVe важен как промежуточный этап

между частотными матрицами и современными нейронными эмбедингами, объединяя статистику корпуса и геометрию смыслового пространства.

Следующий этап развития связан с контекстуальными эмбедингами. В Word2Vec и GloVe слово обычно имеет один вектор независимо от предложения. Но в реальном языке значение слова часто определяется контекстом. Слово «лук» может обозначать растение или оружие, «ключ» – предмет для замка или способ решения задачи, «мышь» – животное или компьютерное устройство. Неконтекстная модель вынуждена смешивать эти значения в одном представлении.

Одним из важных шагов к решению этой проблемы стали глубокие контекстуализированные представления, например ELMo. Питерс подчеркивал, что такие представления моделируют не только синтаксические и семантические свойства слова, но и то, как его употребление меняется в разных языковых контекстах [6, с. 2227–2229]. Это означает, что вектор слова перестает быть постоянной «карточкой» в словаре: он начинает зависеть от предложения, в котором слово находится.

Наиболее известным примером контекстуальных моделей стал BERT. Модель основана на архитектуре Transformer, где ключевую роль играет механизм внимания. Васвани предложил архитектуру, которая отказывается от рекуррентных и сверточных слоев и опирается на attention-механизмы для установления зависимостей между элементами последовательности [7, с. 5998–6001]. Как результат – модель может учитывать, какие слова в предложении особенно важны друг для друга. Контекстуальные эмбединги заметно расширили область применения векторных представлений. Возможность строить векторы не только для отдельных слов, но и для предложений, абзацев, документов появилась именно благодаря прогрессу с BERT.

Однако сила контекстуальных моделей сопровождается усложнением. Если one-hot encoding можно объяснить за минуту, а Word2Vec – через идею соседних слов, то внутреннее представление BERT значительно труднее интерпретировать. Кроме того, такие модели требуют больших корпусов и серьезных вычислительных ресурсов. Но именно они лучше всего отражают живую природу языка: смысл слова не закреплен раз и навсегда, а возникает в конкретном контексте. Именно поэтому контекстуальные эмбединги стали основой современных систем машинного перевода, вопросно-ответных моделей, семантического поиска и больших языковых моделей.

Рассмотрим искусственный язык из 10 слов:

Слово	Значение
kral	король
kvin	королева
man	мужчина
wom	женщина
chi	ребенок
run	бежать
wal	идти
big	большой
sma	маленький
hou	дом

One-hot представление

Слово	Представление
kral	[1,0,0,0,0,0,0,0,0]
kvin	[0,1,0,0,0,0,0,0,0]
man	[0,0,1,0,0,0,0,0,0]
wom	[0,0,0,1,0,0,0,0,0]
chi	[0,0,0,0,1,0,0,0,0]
run	[0,0,0,0,0,1,0,0,0]
wal	[0,0,0,0,0,0,1,0,0]
big	[0,0,0,0,0,0,0,1,0]
sma	[0,0,0,0,0,0,0,0,1]
hou	[0,0,0,0,0,0,0,0,1]

Но несмотря даже на то, что мы формально выразили эти представления в виде числовых массивов, семантически они гораздо корректнее представляются не как «векторы», а как индикаторные кодировки. Каждое слово имеет соответствующую уникальную позицию, а единственная информация, которая в этом представлении кодируется – это лишь факт идентичности слова. И соответственно, по полученным данным никакие выводы о семантических свойствах языка сделать уже невозможно.

В частности, расстояния между любыми двумя словами одинаковы, что исключает возможность выявления сходства; отсутствует группировка слов по категориям (например, «люди», «действия», «признаки»); невозможно восстановить даже простейшие отношения (например, род или иерархию); арифметические операции над такими представлениями не имеют интерпретируемого смысла.

Следовательно, one-hot кодирование не несет семантической информации и не позволяет анализировать структуру языка, выступая лишь в качестве технического способа идентификации токенов.

В качестве конкретного метода рассмотрим модель Word2Vec в архитектуре Skip-gram, обученную на искусственном корпусе предложений, составленных из заданного словаря (например: «kral man», «kvin wom», «man run», «wom wal» и т.д.).

Слово	Вектор (3D)
kral	[0.9, 0.8, 0.1]
kvin	[0.9, 0.7, 0.2]
man	[0.8, 0.6, 0.1]
wom	[0.8, 0.5, 0.2]
chi	[0.7, 0.4, 0.3]
run	[0.1, 0.9, 0.8]
wal	[0.2, 0.8, 0.7]
big	[0.6, 0.2, 0.9]
sma	[0.5, 0.1, 0.8]
hou	[0.3, 0.3, 0.4]

Числовые представления эмбедингов были заданы вручную, но структура их числовых значений повторяет результат обучения модели Word2Vec: слова близкие по смыслу располагаются близко, а различия по семантическим свойствам кодируемы линейными смещениями в пространстве. Данная интерпретация согласуется с наблюдаемыми свойствами распределенных представлений, где семантические отношения и их можно аппроксимировать линейными операциями в векторном пространстве.

В отличие от one-hot кодирования, данные векторы являются плотными представлениями, обученными на основе контекста употребления слов. Наблюдаются следующие свойства: слова автоматически группируются по семантике (персонажи, действия, признаки); расстояния между векторами отражают степень смысловой близости; появляются линейные зависимости между словами. Происходит это, потому что модель извлекает абстрактные отношения (в данном случае – родовые различия) исключительно из распределения контекстов. Даже на упрощенном корпусе видно, что Word2Vec формирует интерпретируемое семантическое пространство.

Дополнительно можно рассмотреть и другие операции с эмбедингами: косинусное сходство. Позволяет количественно оценить близость слов. Например, $\cos(\text{kral}, \text{kvin}) > \cos(\text{kral}, \text{run})$ – отражает их семантическую близость. Усреднение векторов – комбинирование значений слов: $(\text{man} + \text{wom}) / 2 \approx \text{«человек»}$ (абстрактное обобщение);

$(\text{run} + \text{wal}) / 2 \approx$ «движение». Интерполяция – плавный переход между значениями:

$\alpha \cdot \text{man} + (1-\alpha) \cdot \text{woman}$ при $\alpha \in [0,1]$ дает промежуточные представления между «мужчина» и «женщина». Поиск ближайших соседей – для заданного слова можно найти наиболее близкие по расстоянию векторы, что фактически реализует семантический поиск. Кластеризация: группировать векторы можно по группами (допустим, воспользоваться методом k-means), который автоматически выделяет классы слов (люди, действия, признаки).

Исходя из вышеизложенного, эмбединги это не просто цифровое представление, а структура, дающая возможность осуществлять осмысленные геометрические и аналитические операции.

Результаты проведенного анализа показывают, что переход от one-hot кодирования к плотным эмбедингам коренным образом меняет представление языка. Если в первом случае слова являются независимыми символами, то во втором – элементами непрерывного семантического пространства. Эмбединги можно считать мостом между человеческим языком и вычислительной обработкой текста. Они не заменяют человеческого понимания, но позволяют алгоритмам работать со смыслом более гибко, уделяя внимание семантическим признакам, по сравнению с простым кодированием слов.

На упрощенном искусственном примере видно, что современные методы позволяют моделировать смысловые отношения между словами, что на сегодня является собой главный фактор успеха и стандарт современных NLP-систем.

Литература

1. Salton G., Wong A., Yang C. S. A Vector Space Model for Automatic Indexing // Communications of the ACM. 1975. Vol. 18. No. 11. 3. P. 613–620.
2. Harris Z. S. Distributional Structure // Word. 1954. Vol. 10. No. 2–3. P. 146–162.
3. Turney P. D., Pantel P. From Frequency to Meaning: Vector Space Models of Semantics // Journal of Artificial Intelligence Research. 2010. Vol. 37. P. 141–188.
4. Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J. Distributed Representations of Words and Phrases and their Compositionality. In: 2013 Conference on Neural Information Processing Systems. NeurIPS, 2013. P. 3111–3119.
5. Pennington J., Socher R., Manning C. D. GloVe: Global Vectors for Word Representation. In: Conference on Empirical Methods in Natural Language Processing, 2014. Association for Computational Linguistics, 2013. P. 1532–1543.
6. Peters M. E., Neumann M., Iyyer M., Gardner M., et al. 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). Association for Computational Linguistics, 2018. P. 2227–2237.

7. Vaswani A., Shazeer N., Parmar N., et al. Attention Is All You Need. In: 2017 Conference on Neural Information Processing Systems. NeurIPS, 2017. P. 5998–6008.

Embeddings in machine translation systems

R. G. Miftakhova*, A. T. Khanov

*Ufa University of Science and Technology,
32 Zaki Validi st, 450076 Ufa, Republic of Bashkortostan, Russia.*

Email: miftahovar@yandex.ru

The main approaches to the construction of embeddings – vector representations of words that allow machine learning models to take into account semantic and syntactic properties of the language are considered. An overview of key methods, advantages and limitations is presented. An experiment is performed with an artificial language consisting of a small dictionary to identify the differences between one-hot coding and modern embedding methods.

Keywords: machine translation, neural networks, embeddings, vector representation of words.