Concept Embedding for Information Retrieval

Karam Abdulahhad
GESIS - Leibniz Institute for the Social Sciences, Cologne, Germany
karam.abdulahhad@gesis.org

Abstract. Concepts are used to solve the term-mismatch problem. However, we need an effective similarity measure between concepts. Word embedding presents a promising solution. We present in this study three approaches to build concepts vectors based on words vectors. We use a vector-based measure to estimate inter-concepts similarity. Our experiments show promising results. Furthermore, words and concepts become comparable. This could be used to improve conceptual indexing process.

1 Introduction

Conceptual indexing includes the process of annotating raw text by concepts\(^1\) of a particular knowledge source \(^1\). It is used to represent the content of documents and queries by more informative terms, namely concepts rather than words. Annotating text by concepts is used to solve the term-mismatch problem by considering the semantic of text rather than its form \(^1\). For example, the two terms “cancer” and “malignant neoplastic disease” correspond to the same concept (synset) in WordNet\(^2\). However, using concepts instead of words has some side-effects. First, the process of annotating text by concepts is a potential source of noise, e.g. “x-ray” corresponds to more than 6 different concepts in UMLS\(^3\), so which one best fits the original textual content. Second, to better solve the term-mismatch problem, we need to exploit the relations between concepts. Hence, we need a way to quantify these relations. However, inter-concepts similarity is still problematic and non easy to measure \(^1\), because the similarity between two concepts depends on the relation between them. Since relations are different in semantic, e.g. is-a, part-of, etc., and have different properties, e.g. symmetric or not, there is no standard way on how to quantify relations, where it is task-dependent. For example, for one task the is-a relation is much useful than part-of, but for another task the part-of is much useful, and so on.

Word embedding \(^1,12\) has recently proved its effectiveness for several NLP tasks. It is also studied in Information Retrieval (IR), where word embedding is used for ad-hoc retrieval \(^7\), query expansion \(^11,16\), or text similarity \(^8\).

---

\(^1\) Concepts have many definitions \(^1\). A concept here refers to a category ID that encompasses synonymous words and phrases, e.g. UMLS concepts, WordNet synsets.

\(^2\) wordnet.princeton.edu

\(^3\) www.nlm.nih.gov/research/umls/
features make word embedding potentially useful for IR, where a word is a low-
dimensional numerical vector rather than a sequence of characters and algebraic
operations between vectors reflect semantic relatedness between words [10].

Concept embedding takes word embedding to a higher level. It is the process
of representing concepts by low-dimensional vectors of real numbers. Through
concept embedding, one can keep the advantages of both conceptual indexing
and word embedding, and at the same time avoid some of conceptual indexing
disadvantages. More precisely, by using concepts vectors, on the one hand, we
exploit concepts to reduce the term-mismatch effect, and on the other hand, we
avoid complexities related to relation-based inter-concept similarity, and measure
the similarity between two concepts by comparing their corresponding vectors.

In this study, we propose a way to generate concept embedding based on word
embedding. Then, we use concepts vectors in classical IR models. It is worth
mentioning that we do not aim in this study to compare different approaches
of tackling term-mismatch. Hence, we do not report in results any comparison
between our approach and approaches like: pseudo-relevance feedback or word
based expansion. The main goal of this paper is to check the profitability of using
concept embedding, and the adaptability of vector based concept similarity to IR.

2 Related Works

De Vine et al. [5] build medical concept embedding through replacing the tex-
tual content of documents by their corresponding medical concepts, and then
training word2vec [10] on the new corpus, which is now a sequence of concepts.
At the end of the process, they obtain a vector representation for each concept
appeared in the corpus. Choi et al. [2] use a similar approach to obtain concepts
vectors, except that they use temporal information from medical claims to adapt
the definition of context window of word2vec to medical data. Both approaches
build vectors for the concepts that only appear in the corpus and not for all con-
cepts of the corresponding knowledge resources. Furthermore, if we build word
embedding vectors of the same corpus, then concepts vectors and words vectors
will not be comparable, because they are represented in different vector spaces.

Several studies proposed to use word embedding to represent more informa-
tive elements rather than a single word. Clinchant et al. [3] use Fisher kernel to
aggregate words vectors of a document to build a document vector. Le et al. [9]
extend word2vec to be able to compute paragraph-level embedding. Zamani et
al. [13] optimize a query language model to estimate the embedding vector of a
query, where averaging query’s words vectors is a special case of their approach.

Concerning the inter-concepts similarity, many approaches have been used
in literature [11]. They can be categorized [11]: 1- path-based measures, which
depend on the length and the nature of the path that links two concepts within
a knowledge resource; 2- information content measures, which use some corpus-

based statistics to estimate the information content of a concept, and then mea-
suring similarity; and 3- vectors-based measures, which depend on the ability to
represent concepts by vectors, where cos is the main measure in this category.
3 Concept Embedding

We present in this study three methods for concept embedding based on word embedding. The difference between these methods is the additional information that is used, beside word embedding vectors, to build concepts vectors.

**Flat embedding (FEmb):** In this method, we do not use any additional information rather than word embedding vectors. The main hypothesis here is that any concept can be mapped to a set of words. Hence, the embedding vector of a concept $c$ is a function $F$ of the vectors of its words. For example, in WordNet the two words “snake” and “serpent” belong to the “S01729333” synset, so $S01729333 = F(\text{snake}, \text{serpent})$, where $F$ is any function able to merge several vectors in only one vector, e.g. vectors addition, vectors average, etc.

**Hierarchical embedding (HEmb):** Beside word embedding vectors, we use in this method the internal structural information of each concept. This method is initially proposed to deal with UMLS medical concepts, but it is applicable to any resource exhibiting similar concept structure. In UMLS, each concept $c$ consists of several terms, which represent the different forms of text that could be used to express the underlying meaning of $c$, and each term could appear in different lexical variations or strings, where a string can be mapped to either a word or a set of words. Therefore, we have a hierarchy related to each concept. Assume that a concept $c$ consisting of two terms $t_1, t_2$, and each term $t_i$ consists of two strings $s_{i1}, s_{i2}$. In this case, $\vec{c} = F(\vec{s_{11}}, \vec{s_{21}}), F(\vec{s_{12}}, \vec{s_{22}})$.

**Weighted embedding (WEmb):** This method is an extension of FEmb, where we incorporate external statistical information. More precisely, instead of equally treating the words of each concept, we attach a weight indicating their relative importance, i.e. $\vec{c} = F(\alpha_1 \vec{w_1}, \ldots, \alpha_n \vec{w_n})$, where $\alpha_i$ is the weight of $w_i$.

**Evaluation strategy:** Since our goal is to study the profitability of concept embedding for IR, we evaluate the retrieval performance improvement of an IR model that is able to incorporate inter-terms similarity. We use the model of [4]:

$$\text{RSV}(d, q) = \sum_{c \in q} \text{weight}_q(c) \times \text{sim}(c, c^*) \times \text{weight}_d(c^*)$$  \hspace{1cm} (1)

where $\text{sim}(c, c^*)$ is the similarity between two concepts, and $c^*$ is the closest document concept to the query concept $c$ according to the similarity measure $\text{sim}$. If the query concept $c$ also belongs to $d$, then $c^* = c$ and $\text{sim}(c, c^*) = 1$. We use several definitions for $\text{sim}$, some of them are vector based and some are not. By this way we can see if concept embedding vectors are useful for IR.

4 Experimental Setup

**Generating word embedding:** We generate concept embedding vectors based on word embedding vectors. To obtain words vectors, we train word2vec on open access PubMed Central collection, with the following configurations: vector size 500, continuous bag of words, window size 8, and negative sampling is set to 25.

---

[4] www.ncbi.nlm.nih.gov/pmc/, PubMed collection contains: 1177879 vocabularies.
Generating concept embedding: We apply our approach to UMLS2017AA medical concepts, and we only consider the concepts that have English content. Assume the following example for clarification. The concept C0004238 (denoted $c$) has two textual forms or terms: L0004238 (denoted $t_1$) and L0004327 (denoted $t_2$). Term $t_1$ appears in two lexical variations: singular $S0016668$ = “atrial fibrillation” (denoted $s_1$), and plural $S0016669$ = “atrial fibrillations” (denoted $s_2$). The same for term $t_2$ which corresponds to two strings $s_3^2$ and $s_4^2$. By tokenizing, we transform each string $s_j^i$ to a set of words $W_{s_j^i}$ (we remove duplication).

In $FEmb$, the concept vector is: $\bar{c} = \text{avg}(\vec{w}_1^1, \ldots, \vec{w}_l^1)$, where $\vec{w}_i^1$ is the word embedding vector of word $w_i$, $w_i \in \bigcup_{j=1}^{l} W_{s_j^1}$, and $\text{avg}$ returns the average of a set of vectors. For $HEmb$, the concept vector is: $\bar{c} = \text{avg}(\text{avg}(t_1), \ldots, \text{avg}(t_m))$, where $\text{avg}(t_i) = \text{avg}(\text{avg}(s_i^1), \ldots, \text{avg}(s_i^4))$, $\text{avg}(s_i^j) = \text{avg}(\vec{w}_1^j, \ldots, \vec{w}_l^j)$, and $w \in W_{s_j^i}$. Concerning $WEmb$, we follow the same approach as $FEmb$, except that we compute the weighted average $\text{wavg}$ instead of average $\text{avg}$. More precisely, $\bar{c} = \text{wavg}(\vec{w}_1^1, \ldots, \vec{w}_l^1) = \frac{1}{l} \sum \alpha_w \vec{w}_i$, where $l$ is the number of words. The weight $\alpha_w$ of a word $w$ is its $\text{idf}$ score in PubMed, namely $\alpha_w = \ln(N/n)$, where $N$ is the number of documents in PubMed and $n$ is document frequency of $w$.

We generate fixed random vectors for missing words, which means, if a missing word $w$ appears in several concepts, we use the same randomly generated vector. For the $\text{idf}$-weight of missing words, we tested several options: assuming that the word is too popular ($n = N$), too rare ($n = 1$), or in between ($n = \frac{N}{2}$). The three approaches give similar performance; therefore, we only report the first option where $n = N$, which means, a poor $\text{idf}$ score.

Test collections: To evaluate our proposal, we use ad-hoc image-based corpus of ImageCLEF (www.imageclef.org) of years 2011 (clef11) and 2012 (clef12), where documents are captions of medical images with short queries. clef11 has 230K documents and 30 queries. clef12 contains 300K documents and 21 queries (we removed query 14 because it is not mapped to any concept). Documents and queries are mapped to UMLS concepts using MetaMap (metamap.nlm.nih.gov).

IR model and concept similarity: There are three components to be described in the IR model of [1]. The weight of concepts in documents and queries, and the similarity between concepts. To compute the weight of a concept in a document or a query, we apply two classical IR weighting schema: Pivoted Normalization or BM25 [1]. For both models, we use standard parameters values reported in [1]. To compute the similarity between concepts $\text{sim}(c, c')$, we use two measures. The first one is compatible with the vector representation of concepts:

$$\text{sim}(c, c') = \begin{cases} 0 & \cos(\theta) \leq 0 \\ \beta \cos^2(\theta) & \text{otherwise} \end{cases}$$

where $\theta$ is the angle between the two vectors $\vec{c}$ and $\vec{c'}$, and $\beta$ is a tuning parameter. We optimized the value of $\beta$ on clef11 but it is applied to all collections. In our results, we only report the retrieval performance of $\beta = 0.5$. In addition, we only consider the similarity when $\cos(\theta) > 0$, i.e. we ignore the concepts that could have an opposite meaning. We use $\cos^2(\theta)$ instead of $\cos(\theta)$, because it is more discriminant, especially for small angles $\theta \in [-\pi, \pi]$. For comparison,
we use Leacock measure \[11\], which depends on the length of the path of is-a relations between two concepts in UMLS.

5 Evaluation

Table 1 shows results for clef11 and clef12. \(FEmb, HEmb,\) and \(WEmb\) refer to our approaches to build concepts vectors, where the similarity measure between concepts is \[2\]. \(NoEmb\) refer to deal with concepts rather than concepts vectors, where \(Leacock\) refer to the similarity between concepts, whereas, we do not incorporate similarity in \(NoSim\). \(*\) and \(\dagger\) refer to a statistically significant difference with \(NoEmb_{NoSim}\) and \(NoEmb_{Leacock}\), respectively, according to Fisher Randomization test with \((\alpha < 0.05)\).

|                | clef11 | clef12 |
|----------------|--------|--------|
|                | piv    | bm25   | piv    | bm25   |
| NoEmb-NoSim    | 0.1096 | 0.2300 | 0.1552 | 0.3100 |
| NoEmb-Leacock  | 0.1085 | 0.2267 | 0.1505 | 0.2933 |
| \(FEmb_{Eq2}\) | 0.1089 | \textbf{0.2333} | 0.1608 | \textbf{0.3167} |
| \(HEmb_{Eq2}\) | 0.1111 | 0.2100 | 0.1640*\(\dagger\) | 0.3133 |
| \(WEmb_{Eq2}\) | \textbf{0.1137}* | 0.2267 | \textbf{0.1654}*\(\dagger\) | 0.3133 |

Table 1 shows that exploiting relations between concepts and using a relation-based similarity measure introduce noise, where the MAP of \(NoEmb_{Leacock}\) is lower than the MAP of \(NoEmb_{NoSim}\) for both IR models and in both collections. P@10 is also lower in clef11 and slightly better in clef12.

\(WEmb\) gives the best MAP among our approaches, where we use external statistical knowledge beside word embedding vectors. The comparison of \(WEmb\) to \(NoEmb_{NoSim}\) shows that representing concepts by vectors and using vector based inter-concept similarity improve the results. In 3 out of 4 cases the improvement is statistically significant. Moreover, there is no degradation in P@10. If we compare \(WEmb\) with \(NoEmb_{Leacock}\), we see that there is a small gain of MAP (for clef11 and bm25 the gain is statistically significant), and without corrupting P@10. Our approaches to represent concepts by vectors, and use vector-based similarity, improve MAP without corrupting P@10, i.e. the approaches are able to improve results without introducing noise. The only exception is \(HEmb\), where building concepts vectors considers the same word several times if it appears in several strings of the same concept, which represents a possible source of noise.

6 Conclusion

We presented three approaches to build concept embedding vectors based on pre-trained word embedding vectors. We used concepts vectors along with a vector-based similarity to improve IR performance. The results are promising, where the overall performance is improved without losing the absolute precision.
This study can be extended by achieving more in depth free parameters tuning, especially for vector size. Furthermore, we mainly compare the performance of a path-based measure, i.e. Leacock, to a vector-based measure \cite{11}. However, we can also compare the results to content-based measures \cite{11}.

Both words and concepts are represented in the same vector space, so they are comparable. It is thus possible to compare concepts to the original textual content of documents. This is helpful to either achieve conceptual indexing or to improve the quality of some conceptual indexing methods like MetaMap by filtering out non-related or noisy concepts.

References

1. Abdulahhad, K.: Information Retrieval (IR) Modeling by Logic and Lattice. Application to Conceptual IR. Theses, Université de Grenoble (May 2014)
2. Choi, Y., Chiu, C.Y.L., Sontag, D.: Learning low-dimensional representations of medical concepts. AMIA Summits on Translational Science Proceedings 2016, 41 (2016)
3. Clinchant, S., Perronnin, F.: Aggregating continuous word embeddings for information retrieval. In: Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality. pp. 100–109 (2013)
4. Crestani, F.: Exploiting the similarity of non-matching terms at retrieval time. Inf. Retr. 2, 27–47 (February 2000)
5. De Vine, L., Zuecon, G., Koopman, B., Sitbon, L., Bruza, P.: Medical semantic similarity with a neural language model. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 1819–1822. CIKM ’14 (2014)
6. Diaz, F., Mitra, B., Craswell, N.: Query expansion with locally-trained word embeddings. CoRR abs/1605.07891 (2016), \url{http://arxiv.org/abs/1605.07891}
7. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: Semantic matching by non-linear word transportation for information retrieval. In: the 25th ACM International on Conference on Information and Knowledge Management. pp. 701–710. CIKM ’16 (2016)
8. Kenter, T., de Rijke, M.: Short text similarity with word embeddings. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management. pp. 1411–1420. CIKM ’15 (2015)
9. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. pp. II–1188–II–1196. ICML’14, JMLR.org (2014)
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: 26th Inter. Conference on Neural Information Processing Systems. pp. 3111–3119. NIPS’13 (2013)
11. Pedersen, T., Pakhomov, S.V.S., Patwardhan, S., Chute, C.G.: Measures of semantic similarity and relatedness in the biomedical domain. J. of Biomedical Informatics 40(3), 288–299 (Jun 2007)
12. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP. vol. 14, pp. 1532–1543 (2014)
13. Zamani, H., Croft, W.B.: Estimating embedding vectors for queries. In: Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval. pp. 123–132. ICTIR ’16 (2016)
14. Zuecon, G., Koopman, B., Bruza, P., Azzopardi, L.: Integrating and evaluating neural word embeddings in information retrieval. In: Proceedings of the 20th Australasian Document Computing Symposium. pp. 12:1–12:8. ADCS ’15 (2015)