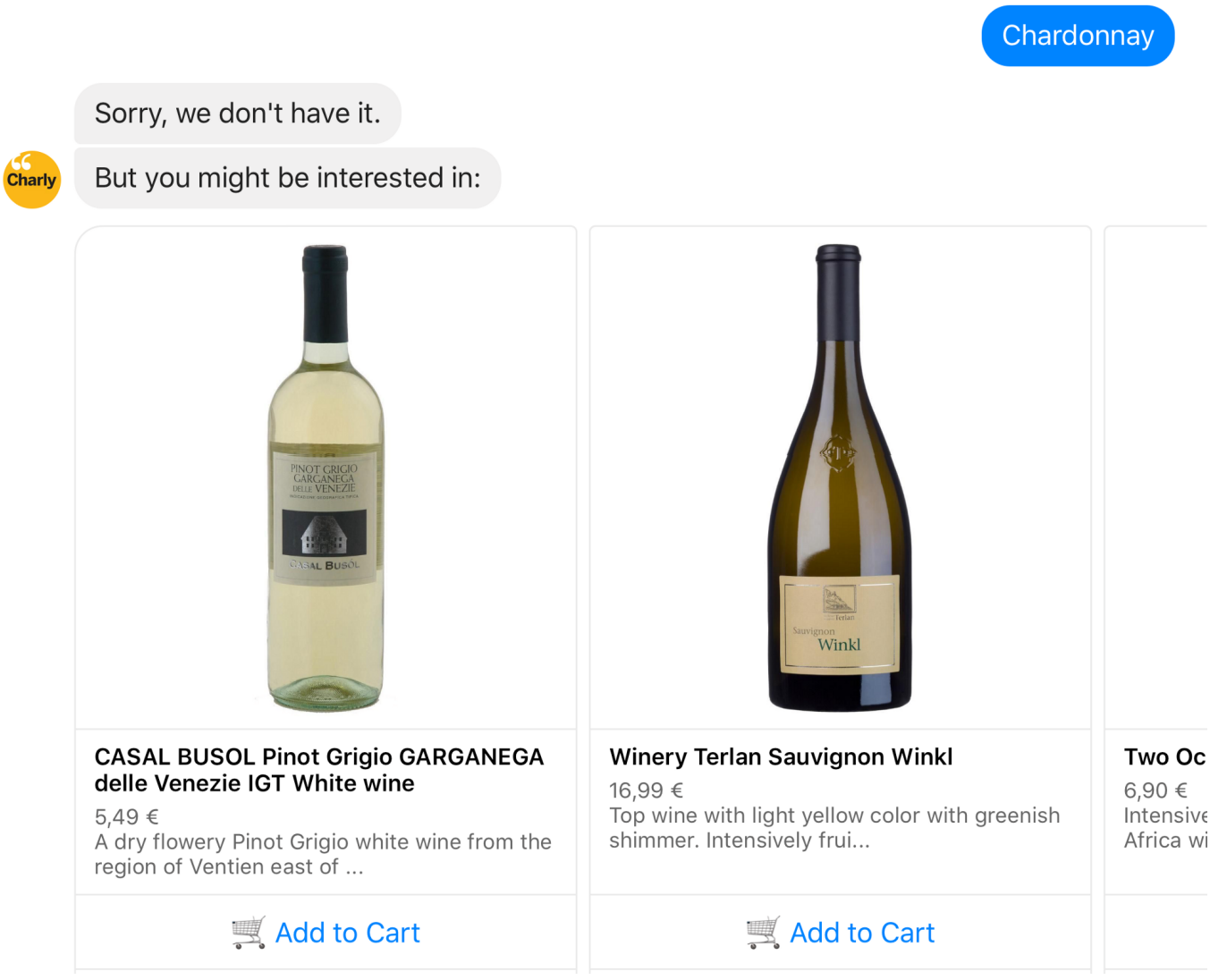
Concept Search by Word Embeddings

Giancarlo Frison Follow
Jun 14, 2018 · 3 min read



original from: https://gfrison.com/2018/06/06/concept-search-by-word-embeddings/

Catalog search is one of the most important factor to the success of e-commerce sites and accurate and relevant results are critical to successful conversion.

The following approach aims to reduce user frustration by presenting related products, when searched items are not available in catalog. The central hypothesis is that an user might buy products with similar characteristics of a product originally searched, leading the successful search into a purchase.

Search engines help to find relevant matches against a query according to various information-retrieval algorithms. Those systems find text occurrences, but regardless their effectiveness, they are unequivocally related to the terms provided by the catalog. Therefore, products cannot be retrieved by words that are not already present in the inventory.

Concept matching (a sub-domain of semantic search) refers to the quality of retrieved instances based on significance. The association of terms by an acceptable grade of relatedness, pivots around those key points:

1. Knowledge gathering. Where is it possible to identify semantic relations among words?

2. Concept extraction. How relations could be extracted and then predicted?

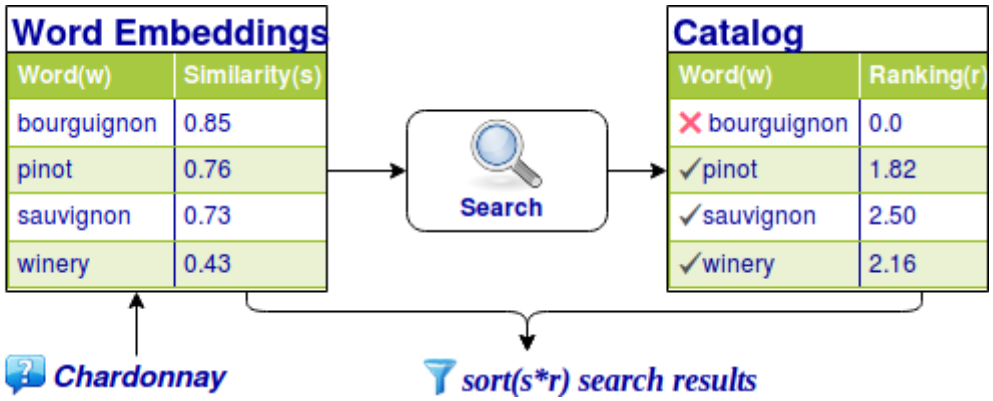
The elaboration applied to the data for obtaining our demanded features is called word embedding.

Word embedding is a very popular term undoubtedly because of the contribution of the deep learning community. It is associate to the research of distributional semantics, the branch of studies for elaborating semantic similarities between words based on their distributional properties.

> "a word is characterized by the company it keeps". cit *R. Firth*

Algorithms (like the well-known skip-gram, cbow, glove) are employed to train models for predict words as they sequentially appears in a given text corpora. As result, the word embedding model converts a single word into a list of similarities, a vector. Analogous words are represented by similar vectors and cosine similarity measures the cosine of the angle between word vectors, thus scoring the relatedness between two words.

Concept Matching Algorithm



In the example above the user submits the unknown search query *Chardonnay* which has some similar terms retrieved in the word embeddings. Some of them might exist in catalog and they are returned to the user.

algorithm *retrieve_alternatives* **is**

input: unrecognized term *query*, word vectors *embeddings*

output: ordered list of products and ranking

query_embeddings ← get similarities of *query* from *embeddings*

results ← empty

for each *w* in *query_embeddings*:

result ←**search by** *w*

result.ranking ←result.ranking * w.score

append result to *results*

return *results* **sort by** ranking

Topic-specific Embeddings

Word embeddings are obtained by elaborating a huge quantity of text, namely *corpus* or *corpora*. There are available several large and structured set of texts for creating word embeddings: Google News corpus, Wikipedia, and so on, as well as word vectors already trained against those corpora. Since the quality of word embeddings reflects the corpus

from which it has been generated, I purposely created a topic-specific corpora specialized in food, by scanning more than **600** food blogs and collecting roughly **40 Mb** of prepared text. The amount of text is risible in comparison with Google News but nonetheless it is enough for the purposes of computing similarity in the small range of catalog queries. The preparation of corpora includes the remotion of everything but words, case conversion and sentence tokenization. I choose fastText for elaborating text representations, it uses sub-word information to build vectors for unknown words and as the name might suggest, it is really fast.

This solution has been filed as *"System, computer-implemented method and computer program product for information retrieval"* at the European patent office. It is applicable to many different domains, like in clothing, automobile, electronics retail, just by getting the proper specialized corpora from which word similarity can be inferred.

)

Machine Learning Word2vec Deep Learning Search Information Retrieval