Sentence Transformers for Document Similarity

excerpt from the upcoming article "The Employment Impact of Emerging Digital Technologies"

Ekaterina Prytkova, PhD

Embeddings with Sentence Transformer

We measure industrial and occupational exposure to the core digital technologies as *textual similarity* between the description of the innovation (from the patent) and industrial/occupational description. To leverage patent, industrial, and occupational descriptions for exposure estimation, we first need to represent these texts in a numerically meaningful form. To achieve this, we employ the state-of-the-art NLP techniques. We encode the raw text of each document into numerical representation using a pre-trained language model; such n-dimensional dense vector representation of text is called *embedding*.

A variety of techniques can produce vector representations of text: from the simplest frequency-based methods, such as TF-IDF, to bag-of-words (BoW) models, such as Word2Vec or FastText, and seq2seq models, such as recurrent neural networks (RNNs), to various transformers. We choose the all-mpnet-base-v2 sentence transformer for two crucial reasons.

First, the all-mpnet-base-v2 transformer produces *contextual or dynamic embeddings*: encoding of a word's meaning that accounts for its surrounding context; the same word in different documents has a different vector that encodes its semantic meaning. Therefore, the content of the entire document is represented in greater detail than using BoW models that produce static embeddings, i.e., fixed vectors for a word in all documents. Encoded variety of semantic meanings per word and resulting contextual representation of the entire document in dynamic embeddings is a crucial advantage over static embeddings in tasks that involve cross-domain document matching, such as similarity between technology and industry/occupation. The frequency-based methods that represent documents as (weighted) word-count vectors are superseded by embeddings-based methods for a number of substantial reasons; for example, they cannot handle new words (fixed vocabulary), struggle with negation and polysemy, and require exact matching leading to sparse representation, etc. Therefore, matching documents represented by word-count vectors—especially between two *different* corpora—is substantially less robust than using contextual embeddings.

Second, even among models that produce contextual embeddings, the all-mpnet-base-v2 transformer has a significant advantage: it has been trained using *contrastive learning*. During the training process, the all-mpnet-base-v2 transformer is given triplets of documents, each consisting of a focal document, one similar document (positive example), and one dissimilar document (negative example). The model's objective is to learn such document representations that similar documents have closely distanced embeddings and dissimilar documents have embeddings positioned far apart. Thus, contrastive learning explicitly ensures that *distances* between document-vectors represent their semantic (dis)similarity. Other models trained with alternative procedures, e.g., BERT with masked language modeling, produce nuanced document representations but *distances* between them are not accurate representations of semantic similarity between documents.[1]

In sum, all-mpnet-base-v2 produces both nuanced document representations (contextual embeddings), and distances between them accurately reflect semantic similarity.

Digital Technologies

We propose to represent digital *technologies as clusters of patent embeddings*. For each patent title p, we obtain its embedding Emb_p. We then cluster the embeddings using the k-means algorithm to obtain 40 clusters, which we designate as our set of digital technologies $\mathcal{K}$.

Initially, we compute partitions ranging from 5 to 100 clusters and record each Davies-Bouldin Index (DBI) score. The optimal range, based on the lowest DBI scores, lies between 30 and 45 clusters,

[1] In technical terms, the resulting embedding space is anisotropic, i.e. non-uniform in different directions, and frequent words concentrate densely while rare words are sparsely scattered; this problem propagates with pooling further to document level.

Table 1: List of Digital Technologies

	Family		Emerging Digital Technology
F1	3D Printing	01	3D Printer Hardware
		02	3D Printing
		03	Additive Manufacturing
F2	Embedded Systems	04	Smart Agriculture & Water Management
		05	Internet of Things (IoT)
		06	Predictive Energy Management and Distribution
		07	Industrial Automation & Robot Control
		08	Remote Monitoring & Control Systems
		09	Smart Home & Intelligent Household Control
F3	Smart Mobility	10	Intelligent Logistics
		11	Autonomous Vehicles & UAVs
		12	Parking and Vehicle Space Management
		13	Vehicle Telematics & Electric Vehicle Management
		14	Passenger Transportation
F4	Food Services	15	Food Ordering & Vending Systems
F5	E-Commerce	16	Digital Advertising
		17	Electronic Trading and Auctions
		18	Online Shopping Platforms
		19	E-Coupons & Promotion Management
F6	Payment Systems	20	Electronic Payments & Financial Transactions
		21	Mobile Payments
		22	Gaming & Wagering Systems
F7	Digital Services	23	Digital Authentication
		24	E-Learning
		25	Location-Based Services & Tracking
		26	Voice Communication
		27	Electronic Messaging
		28	Workflow Management
		29	Cloud Storage & Data Security
		30	Information Processing
		31	Cloud Computing
		32	Recommender Systems
		33	Social Networking & Media Platforms
		34	Digital Media Content
F8	Computer Vision	35	Augmented and Virtual Reality (AR/VR)
		36	Machine Learning & Neural Networks
		37	Medical Imaging & Image Processing
F9	HealthTech	38	Health Monitoring
		39	Medical Information
		40	E-Healthcare

Notes: This table lists the 40 digital technologies along with their respective technology families. These digital technologies are obtained by clustering the embeddings using the k–means algorithm, where the embeddings are derived with the sentence transformer all-mpnet-base-v2. Technologies are grouped by families, where a family comprises technologies whose occupation structure of semantic links is highly correlated.

indicating high within-cluster and low between-cluster similarity. We further examine these partitions by using the most representative phrases per cluster via c-TF-IDF, where 'c' stands for *class*, or in our case *cluster*. Human comprehension of clusters' content summarized in representative phrases helps determine the optimal number of clusters from a prospective data-driven range.

We find that 40 clusters are optimal for our analysis, as they align well with commonly discussed technologies in digital and automation literature. Table 1 presents our set of prominent digital technologies grouped by technology families.