Can we Predict Innovation? Narrow Experts Versus Competent Generalists

Amir Hazem
RCAST - The University of Tokyo
amir.hazem@gmail.com

Kazuyuki Motohashi
RCAST - The University of Tokyo
motohashi@tmi.t.u-tokyo.ac.jp

Abstract

In this paper, we investigate the role of large language models in predicting innovation. We contrast two main paradigms: i) narrow experts: which consists of supervised and semi-supervised models trained or fine-tuned on a specific task and ii) competent generalists: which consists of large language models with zero-shot and few-shots learning. We define the task of innovation modeling and present the first attempt to understand the transformation from research to innovation. We focus on product innovation which can be defined as the process of transforming technology to a product or service and bring it to the market. Our extensive empirical evaluation shows that most existing pretrained models are not suited and perform poorly on the innovation modeling task. We also show that injecting research information helps improving the alignment from technology to the market. Finally, we propose a new methodology and fine-tuning strategies that achieve significant performance boosts over the baselines[1]

1 Introduction

In order to stay competitive, companies have to be innovative and are in perpetual need of new ways to improve and extend their product portfolio. Understanding innovation process is a challenge for industries and scholars, as it is viewed as a catalyst for competitiveness (Kline and Rosenberg, 2009). Innovation is commonly defined as the introduction of something new (idea, method or device) or a change made to an existing product or field (Kahn, 2018). Invention represents the first occurrence of a new idea[2] while innovation is about the commercialization of this idea, i,e. bringing it to the market (Kalogeras and Anagnostopoulos, 2012).

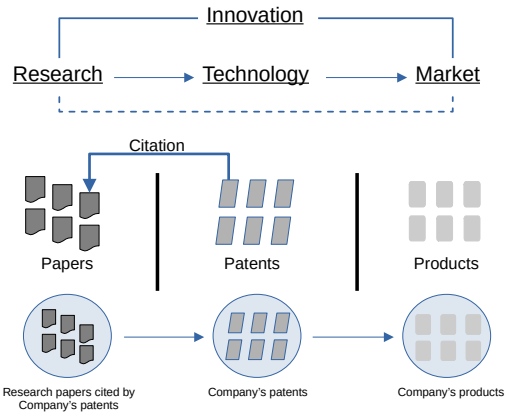

Figure 1: Innovation representation from research and technology to the market

Figure 1 illustrates the innovation workflow in which company's technology is represented by its granted patents and company's market is represented by its commercialized products. The interaction between technology and the market is reflected by the patents that have led to the release of new commercial products. While the interaction between technology and research captures the extent to which patents are based on scientific knowledge of the cited academic papers. Finally, the interaction between research and the market reflects a direct link between science and industry bypassing technology (i.e. no granted patent is available for a released product) [3].

The main purpose of the innovation modeling task is to align, for a given company, its technology and its market. i, e. matching companies' granted patents with their corresponding products. It is not rare that many patents remain unused or can be applied in different fields to produce new products.

[1]Our dataset, code and appendices are available here: https://github.com/AmirHazem/Prim

[2]We adopt the admitted definition of innovation in the business and management field and make a clear distinction between ideation which consists in elaborating new ideas, from the innovation process which starts from the validation of an idea as a technology (granted patent) to its use on the market as a new product.

[3]In this paper we investigate both direct and indirect links between research papers and commercialized products.

Proceedings of Recent Advances in Natural Language Processing,pages 413–422
Varna, Sep 8–10, 2025
https://doi.org/10.26615/978-954-452-098-4-050

A direct application of innovation modeling is to build a recommendation system for patents without specific commercial products or by extending their application to other domains.

If innovation modeling has been extensively studied in the business and management fields (Kahn, 2018), it is still a recent and rarely addressed task for the nlp community[4]. To the best of our knowledge, there has been no established benchmark or comprehensive empirical evaluation of the role of research in innovation that has been conducted so far. In this work, we leverage innovation modeling and establish new baselines that can be used for future research in innovation modeling and prediction.

Innovation is essentially driven by science and technology. If the role of technology in innovation can be derived from the relation between its granted patents and the released products, it is unclear how research contributes to this process. In this work, we first evaluate innovation modeling under its primary workflow which consists in the transformation from technology to the market. Then, we investigate the role of research papers in the transformation process based on the patent to paper citation information and propose new strategies to improve the alignment performance. More specifically, we investigate the role of large language models in predicting innovation. We contrast two main paradigms: i) narrow experts: which consists of supervised and semi-supervised models trained or fine-tuned on a specific task and ii) competent generalists (Radford et al., 2019): which consists of LLMs with zero-shot and few-shots learning.

Our first contribution is to propose an evaluation framework that includes a benchmark and a dataset for innovation modeling. The dataset comprises: (i) a set of 1,638 stock market companies; (ii) a technology corpus of 60,722 English granted patents; and (iii) a market corpus represented by companies' products. In addition, we introduce (iv) a scientific corpus comprising 65,356 research papers cited by the granted patents of the technology corpus. Our second contribution is an extensive evaluation of LLMs and the proposition of a simple but yet effective fine-tuning method that combines science and technology to match its corresponding products on the market.

[4]Two works we are aware of, that addressed technology with no empirical evaluation and no incorporation of research in innovation Lee et al. (2020); Motohashi and Zhu (2023).

2 Related Work

The invention of the perceptron (Rosenblatt, 1958) showed that for the first time, a machine could perform a given task based on some examples. This paradigm, reinforced by the introduction of backpropagation (Rumelhart et al., 1986; Lecun, 1986) certainly marked the beginning of the long journey of neural networks from its birth with the perceptron to the nowadays large language models. Later on, the development of machine capacity allowed to train powerful models such as RNNs LSTMs, and CNNs on a specific task. The common paradigm of all these models is that they were meant to be narrow experts. Which means trained for a specific task. The introduction of the transformer (Vaswani et al., 2017) however, marked another shift in paradigm. Instead of training a model for each specific task, transformers have shown that it is possible to train a model for multiple tasks. Eventually, this model can be fine-tuned for a downstream task. Nowadays, LLMs are meant to be competent generalists (Radford et al., 2019).

Large language models (LLMs) are transformer-based generative models trained on massive text collections (Vaswani et al., 2017). They demonstrate remarkable accuracy and generalization ability when fine-tuned on downstream applications (Devlin et al., 2019). They are also capable zero- and few-shot learners, able to generalize to tasks unseen during training (Brown et al., 2020). The Transformer (Vaswani et al., 2017) is the foundational architecture for most modern LLMs including ChatGPT. It is solely based on attention mechanisms and is composed of an encoder of bidirectional attention blocks and a decoder of unidirectional attention blocks. LLMs can be grouped into three categories: 1) encoder-only LLMs such as BERT (Devlin et al., 2019), 2) decoder-only LLMs such as GPT family (Generative Pretrained Transformer) (Radford et al., 2018, 2019; Brown et al., 2020) and 3) encoder-decoder LLMs such as T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020) and BART (Lewis et al., 2020) (Bidirectional and AutoRegressive Transformers).

Despite the performance of encoder-only and encoder-decoder models, they have the downside that requires significant amount of task-specific data for fine-tuning. GPT-3 (Brown et al., 2020), a decoder-only based model has demonstrated that extremely large autoregressive language models can be used for few-shot predictions, where the

model is only given a natural language task description and optionally a handful of examples (few-shots) demonstrating how the task should be performed. Nowadays, LLMs have predominantly adopted the decoder-based architecture and a number of powerful models have been developed such as PaLM (Chowdhery et al., 2022), Galactica (Taylor et al., 2022) and GPT-4 (Achiam et al., 2024).

The performance of LLMs scales predictably as a power-law with the number of parameters (Kaplan et al., 2020). This outcome has led to an abundance of research focusing on scaling Transformer models up to ever-larger scales that surpass 500B parameters (Smith et al., 2022). The overall improvements in LLMs come from: i) scaling the size of the models in both depth and width; (ii) increasing the number of tokens that the model is trained on; and (iii) training the model on cleaner datasets and from diverse sources (Chowdhery et al., 2022).

Addressing a specific task using LLMs follows two paradigms: 1) Fine-tuning and 2) Prompt-based learning (Brown et al., 2020). Fine-tuning has been the most common approach and involves updating the weights of a pretrained model by training on a supervised dataset (Devlin et al., 2019; Raffel et al., 2020). If this approach achieves strong performance, it is task-dependent and exhibits poor generalization. Furthermore, it can not be applied to the very recent LLMs with billions of parameters[5]. Prompt-based learning also called in-context learning via zero-shot, one-shot or few-shot prompting (Brown et al., 2020), is more effective in leveraging the knowledge encoded in LLMs and only requires few shots learning. It consists of adding natural language text like short phrases to the input or output to encourage the pretrained model to perform a specific task. Prompting does not require updates to LLM's parameters reducing computational requirements as compared to fine-tuning approaches.

Applying LLMs to innovation modeling can be addressed following the two above described paradigms (fine-tuning and prompting) under certain conditions. While most recent LLMs are task agnostic, we consider innovation modeling as a specific task and expect fine-tuning to be more appropriate than a general purpose model. If LLMs with billions of parameters can not be fine-tuned, the increase of open-source LLMs that provide smaller versions of the same model with moderate number of parameters (few millions) such as OPT (Zhang et al., 2022) and MoE (Artetxe et al., 2022), make it possible to perform fine-tuning if needed.

3 Technology to Market Alignment Task

We address innovation modeling as a sentence similarity task where technology and market embeddings are matched using the cosine similarity (Hazem et al., 2024). The representation of technology is based on patent abstracts[6] while the market representation is based on product description. Each company C_i ($i \in [1, n]$ where n in the number of companies) is represented by a technology/market pair (C_i^t, C_i^m) where C_i^t represents the technology of the company and C_i^m represents its market. Hence, $C_i^t = \{pat_1, pat_2, ..., pat_m\}$ where pat_1 for instance is a given patent of the company C_i and $C_i^m = \{prod_1, prod_2, ..., prod_k\}$ where $prod_1$ is its corresponding market product.

The alignment of technology and market of companies is conducted as follows: 1) compute the technology centroid embedding of all the companies (cf equation1); 2) compute their market centroid embedding (similarly to equation 1), and 3) rank the market candidates for each technology according to the cosine similarity.

$$C_i^t = \frac{1}{p}\sum_{j=1}^{p}(Emb(pat_j)) \quad (1)$$

p is the number of patents of C_i^t. To compute the embedding of a given patent, we replace Emb of equation 1 by any sentence embedding model. When using SBERT (Reimers and Gurevych, 2019) for instance, the centroid computation of technology is given by:

$$C_i^t = \frac{1}{p}\sum_{j=1}^{p}(SBERT(pat_j)) \quad (2)$$

4 Proposed Method

To incorporate research into the technology to market alignment, we investigate three combination strategies: 1) direct embedding combination; 2) combination during fine-tuning and 3) Eventually, the fine-tuned embeddings obtained from the fine-tuning combination, can be used for direct embedding combination as defined in the first strategy.

[5]Fine-tuning alternatives exist such as bitfit (Ben Zaken et al., 2022) and Lora (Hu et al., 2022).

[6]Patent abstract is a common use to represent technology in the literature (Lee et al., 2020) as it contains the main information about the developed technology.

In the following sections, we describe the three proposed strategies.

4.1 Embedding Combination

A company usually holds several patents. Eventually, each patent can cite one or several research papers. Given the technology centroid embedding of the company C_i^t, and its research centroid representation (noted C_i^r), we simply average the two embeddings and obtain the combined technology and research embedding vector C_i^{rt} as follows:

$$C_i^{rt} = \frac{1}{2}(C_i^t + C_i^r) \quad (3)$$

We assume that the two embedding representations (C_i^t and C_i^r) can be complementary and evaluate this straightforward but yet necessary approach in our experiments.

4.2 Combination during Fine-Tuning

To improve the embedding representation of technology and market pairs of each company, we use SBERT architecture at inference (Figure 2). We take advantage of the siamese network and fine-tune SBERT. Our training data comprises triplets of (patent, paper, product). To encoded the direct link (signal) between each pair of the innovation workflow, we use the Multiple Negative Ranking Loss (Henderson et al., 2017) of sentence-transformers[7] represented as (anchor, positive, negative), we generate for each training triplet, 3 inputs as illustrated in Table 1.

Triplet Input		
Anchor	Positive	Negative
patent	paper	paper
patent	product	product
paper	product	product

Table 1 Input representation for SBERT fine-tuning.

Our model is fine-tuned with positive input pairs of patents and cited papers or products as well as randomly selected negative pairs[8].

4.3 Combination after Fine-Tuning

Our third and final strategy is to combine patent and research paper embeddings as described in section 4.1. However instead of using existing pretrained models, we use our fine-tuned model as described in section 4.2. This can be represented in the following equation:

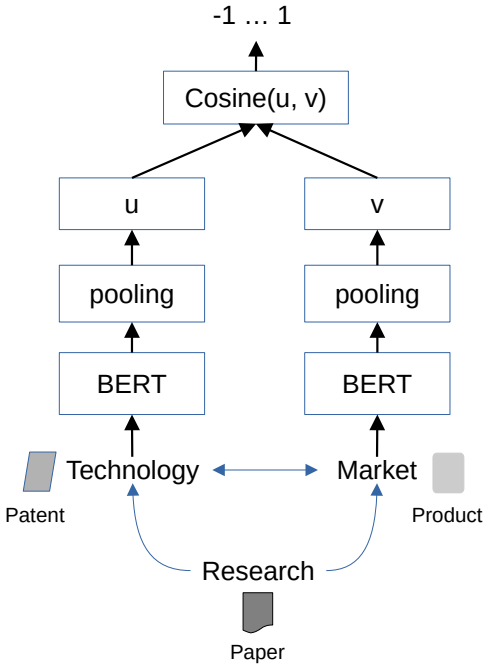

Figure 2: SBERT architecture using research, technology and market information as inputs for fine-tuning.

$$C_i^{rt} = \frac{1}{2}(\frac{1}{p}\sum_{j=1}^{p} FT(pat_j^t) + \frac{1}{k}\sum_{j=1}^{k} FT(pap_j^r)) \quad (4)$$

where FT is our fine-tuned model; p is the number of patents (pat^t) held by company i and k is the number of the corresponding papers (pap^r) cited by p patents. It is to note that our method is not limited to SBERT. Any compatible transformer model can be used. We report in our experiments several other pretrained models (as shown in Table 6).

5 Experiments

To comprehend the role of research in innovation, we consider 3 scenarios: i) technology to market alignment; ii) research to market alignment; and finally iii) the combination of research and technology to market alignment. The combination is addressed in 3 ways: 1) direct embedding combination, 2) combination during fine-tuning and 3) direct embedding combination after fine-tuning.

[7] https://sbert.net/docs/package_reference/sentence_transformer/losses.html#multiplenegativesrankingloss

[8] We let the exploration of hard negatives for future work.

5.1 Dataset

The innovation modeling dataset comprises: (i) 1,638 stock market companies extracted from Crunchbase[9] database; (ii) a technology corpus of 60,722 English granted patents extracted from the United States Patent and Trademark Office (USPTO)[10]; and (iii) a market corpus represented by companies' products description extracted from Crunchbase. In addition, we use (iv) a scientific corpus comprising 65,356 research papers cited by the granted patents[11]. In order to evaluate the impact of research in innovation, we only keep companies that have both patents and research paper citations. Overall, 990 companies out of 1,638 have patents that cite papers. Table 2 shows the training, development and test size.

	train	dev	test
Company	590	100	300
Patent	6,691	797	3,493
Paper	47,927	5,036	16,983

Table 2 Dataset evaluation statistics.

Table 3 shows patents distribution per company. On average, a company has approximately 26 patents. The companies with a maximum patents of 2,502 is *vmware* and a minimum patent of 1 is accolade for instance[12]. The total number of cited papers is 65,356 papers and the total number of patents is 10,981 patents as shown in Table 4.

	Min	Max	μ	σ	#Total
Patent	1	2,502	26.80	21.23	43,900

Table 3 USPTO granted patent statistics per company (out of 1,638 companies)

5.2 Baseline Methods

We evaluate sentence-level and document-level embedding representations as well as generative LLMS. As sentence level representation we evaluate: Sentence-T5 (Ni et al., 2022), Sentence-BERT (Reimers and Gurevych, 2019) and e5 (Wang et al., 2022). As document level representation, we evaluate, Specter (Cohan et al., 2020), a method which generates document-level embedding of scientific documents based on the pretrained transformer SciBERT. Specter encode the citation information between signals. We consider this method as a strong baseline as it is fine-tuned on paper citation pairs. We also use the recent Specter2 version (Singh et al., 2022). Furthermore, we evaluate Instructor (Su et al., 2022), an instruction based model for embedding representation which was fine-tuned on 300 tasks including science representation. As generative models, we experiment with Llama3.1 (8B) (Touvron et al., 2023) and Gemma2 (9B) (Gemma, 2024) with zero-shot and few-shots learning. Our prompts consist of asking the model to provide a similarity score between -1 and 1 for a pair of input sentences (Patent and product for instance). In the few-shots learning configuration we provide pairs of positive and negative examples[13]. Finally, we evaluate Sentence-GPT (SGPT) (Muennighoff, 2022), a GPT-based sentence embedding model for semantic search.

	Min	Max	μ	σ	$\#Total$
	Number of patents per organization				
Patent	1	310	11.09	7.74	10,981
	Number of cited papers by organization				
Paper	1	14,792	282.74	58.74	65,356
	Number of patents citing the same paper				
Patent	1	304	4.28	3.50	10,981
	Number of cited papers per patent				
Paper	1	495	25.49	10.27	65,356

Table 4 Data distribution statistics for patents and cited papers. #Total is number of patents for all the companies and for the total number of cited papers.

6 Results

Table 5 reports the results of existing pretrained sentence-level embedding models, as well as document-level and generative LLMs.

Overall, we observe that the best performing model is the document-level approach Specter (Map = 48.07) which was fine-tuned on research paper citations. The second best performing model is e5-7B-instruct (Map = 47.27) which was fine-tuned on a curated dataset and initialized with Bert-

[9] Crunchbase provides information about private/public companies: https://www.crunchbase.com/

[10] https://patentsview.org/download/data-download-tables

[11] Marx and Fuegi (2020) describe the procedure to get the citation information.

[12] Search engine systems for matching medical providers and patients. This patent was granted in 2020-12-08.

[13] We varied the number of shots from one to six but did not observe significant difference in the results.

	Tech→Market			
Model	**Ac1**	**Ac5**	**Ac10**	**MAP**
ST5-Base	21.00	39.67	46.67	30.02
ST5-Large	22.00	39.67	48.33	31.23
ST5-XL	25.33	44.00	54.00	34.79
ST5-XXL	27.00	47.33	57.00	37.21
SBERT-Base	09.00	23.67	30.33	17.01
SBERT-Large	10.33	26.33	38.33	18.96
SBERT-MiniLM	32.00	51.00	60.33	41.62
SBERT-MPNet	32.67	57.67	66.33	44.67
e5-Base	27.67	48.00	58.33	38.19
e5-Large	25.67	47.67	54.00	36.64
e5-7B-instruct	36.00	60.33	70.00	47.27
Specter	**36.33**	**61.33**	**72.00**	**48.07**
Specter2	36.00	57.67	70.33	46.97
SGPT-125M	15.67	29.00	37.00	23.08
SGPT-1B	19.00	36.67	44.67	27.95
SGPT-5B	21.00	43.33	56.00	31.63
Instructor-Base	22.00	44.00	49.00	31.71
Instructor-Large	29.33	55.00	66.67	41.68
Instructor-XL	27.00	52.67	62.33	38.56
Llama3.1 (0-shot)	27.33	49.00	59.00	37.69
Llama3.1 (n-shots)	29.66	43.66	48.66	36.09
Gemma2 (0-shot)	33.00	52.33	60.00	42.21
Gemma2 (n-shots)	29.33	55.33	64.33	41.78

Table 5 LLM's performance on the innovation (technology to market) alignment task (Test set). The performance is measured in terms of accuracy (Ac@1, 5 and 10) and Mean Average Precision (MAP%)).

MiniLM. It is to note that e5-base and e5-large perform poorly, which suggests that e5 effectiveness is due to scaling with instruction learning. However, we remark that the Instructor model, including its very large (XL) version also performs poorly. This suggest that this generalist model which was trained on 300 tasks can not adapt to our specific task. We note the same observation for the two generative models Llama3.1 and Gemma2. The third best performing model after Specter2 is SBert-MPNet followed by Gemma2. Finally, sentence-T5 (ST5) which is based on the T5 encoder-decoder fails to provide efficient representation of technology and market even by scaling with the ST5-XXL. This may suggest that the initialization of ST5 with encoder-decoder word representation such as T5 is not suited, at least in our task. We finally observe that SGPT which uses GPT transformer to represent sentences, also failed in providing a good innovation prediction, even at scale. This suggests that decoder-only based LLMs (unless fine-tuned) can not be applied to innovation modeling where the relation between technology and market does not solely rely on semantics but there must be some underlying signal that can not be captured only from a general purpose sentence pair training.

We report in Table 6 the results of selected baselines as well as our proposed strategies. We first observe that most of the tested baselines show improvements when only combining technology and research embeddings. For instance among the baselines, SBERT-MPNet obtained the best combination score (Map = 51.79), closely followed by Specter (Map = 50.17). It is also interesting to notice that Specter performs better than SBERT-MPNet when no combination is performed (Map = 48.07 for Specter vs Map = 44.67 for SBERT-MPNet). It is also worth noticing that embedding combination does not always improve the alignment score. For instance Instructor-Large obtains almost no gain while e5-base and SBERT-base performance drops.

The second part of Table 6 shows the obtained results after fine-tuning our models. We fine-tuned using SBERT-base, e5-base, and SBERT-MiniLM. We observe that our models exhibit improvements before and after embedding combination where pretrained models such as e5 and SBERT-Base failed. This finding suggests that our methodology of injecting research by first fine-tuning and then embedding combination is effective to improve innovation modeling. We see for instance a boost by a large margin of our fine-tuned model with SBERT-MiniLM (Ours (SBERT-MiniLM)) before embedding combination with an increase from MAP = 41.62 to 50.92 for the technology to market alignment for instance. After embedding combination, this model achieves a Map score of 52.30, while it only achieved 42.57 before fine-tuning. Our best model[14] obtains 55.02 of MAP.

7 Ablation Study

We investigate the impact of different loss functions and fine-tuning inputs. All the reported results in this section are based on the pretrained SBERT-

[14] SBERT-MPNet fine-tuned using the entire training data with research papers only.

	Baselines								Proposed Combination			
	Tech→Market				Res→Market				Tech+Res→Market			
Model	Ac1	Ac5	Ac10	MAP	Ac1	Ac5	Ac10	MAP	Ac1	Ac5	Ac10	MAP
SBERT-Base	09.00	23.67	30.33	17.01	04.33	13.00	20.67	10.33	08.67	22.33	29.67	15.66 ↓
SBERT-Large	10.33	26.33	38.33	18.96	06.33	19.33	26.67	13.67	11.00	25.00	37.67	19.33↑
SBERT-MiniLM	32.00	51.00	60.33	41.62	25.67	45.33	54.00	35.47	32.00	52.0	66.00	42.57 ↑
SBERT-MPNet	32.67	57.67	66.33	44.67	29.67	56.33	64.67	42.22	39.67	64.33	72.33	51.79 ↑
e5-base	27.67	48.00	58.33	38.19	17.33	35.00	42.67	26.20	27.67	46.67	54.67	37.05 ↓
Specter	36.33	61.33	72.00	48.07	28.00	58.67	68.67	41.02	37.33	65.67	76.00	50.17 ↑
Specter2	36.00	57.67	70.33	46.97	32.00	**59.00**	69.00	44.08	35.67	64.67	74.67	48.53 ↑
Instructor-Large	29.33	55.00	66.67	41.68	22.00	43.00	55.33	33.42	30.00	55.00	67.33	41.80↑
Proposed Fine-tuning									**+ Embedding Combination**			
Ours (SBERT-base)	24.33	50.33	60.33	36.18	20.00	44.00	55.67	31.63	25.67	51.67	62.67	37.96 ↑
Ours (e5-base)	33.67	56.33	63.67	44.62	30.00	49.33	62.00	40.08	37.67	58.67	69.67	48.33 ↑
Ours (SBERT-MPNet)	39.67	59.33	71.00	49.61	32.67	55.67	66.67	43.75	41.00	65.00	74.33	52.14 ↑
Ours (SBERT-MiniLM)	39.67	64.33	75.33	50.92	30.67	57.33	69.33	43.23	40.67	65.33	77.67	52.30 ↑
Ours (BEST)	**42.00**	**69.33**	**77.00**	**53.95**	**32.67**	58.33	**73.33**	**44.79**	**41.67**	**72.33**	**78.67**	**55.02** ↑

Table 6 Evaluation on the test set of baselines and our proposed combination strategies on the innovation alignment task (Accuracy (Ac@1, 5 and 10) and Mean Average Precision (MAP%)).

MiniLM model that we fine-tuned.

	Tech→Market			
Loss	**Ac1**	**Ac5**	**Ac10**	**MAP**
MNRank	46.0	78.0	86.0	60.45
Cosine	29.0	68.0	79.0	45.44
Contrastive	32.0	64.0	81.0	47.32

Table 7 Development results with different Loss functions. MNRank: Multiple Negative Ranking Loss.

In Table 7, we contrast the Multiple Negative Ranking Loss with the Cosine loss and the Contrastive Loss. We observe that both Cosine and the Contrastive loss performs poorly while compared to the Multiple Negative Ranking loss.

	Tech→Market			
Model	**Ac1**	**Ac5**	**Ac10**	**MAP**
MNRank	46.0	78.0	86.0	60.45
-product	37.0	70.0	82.0	52.11
-paper	49.0	77.0	84.0	62.53
-patent	52.0	78.0	87.0	63.12

Table 8 Development results removing patent, product or paper from the training inputs. This results in fine-tuning using (anchor, positive) pairs by leaving one out for each ablation experiment.

Table 8 shows that removing product information from the fine-tuning process results in a significant decrease in performance. Surprisingly, removing patents or papers leads to some improvement in MAP score but only due to an improve in accuracy at 1 (Ac1) which has a higher impact on the MAP score. This behavior can be explained by the overlap that may exhibit some patents and papers. Further investigations on the proximity and overlap between patents and products should be carefully addressed for better representation. We let this investigation for our future work.

8 Discussion

As LLMs are trained on a huge amount of data of different domains, it is reasonable to question if there is still a need for fine-tuning or better follow the new trend of prompt-based approaches. Another question is: whether it is necessary to use huge general purpose LLMs for a task that can be addressed with smaller models adapted for a downstream application? In innovation modeling, the underlying link between patents and there corresponding products represent a strong signal that may not be captured only by general semantic training. Models such as SGPT for instance, which is trained for semantic search failed to predict innovation. To better apprehend LLMs performance, it is necessary to have a closer look at there architecture. If LLMs are based on the transformer, each model has its own specificity. In the following we will discus architecture similarities and differences.

Existing LLMs are based on encoder-only (such

as BERT), encoder-decoder (such as T5) and decoder only architecture (such as GPT3 and Llama). In our experiments, sentence-based models (such as SBERT) based on the Siamese BERT-Networks showed the best performance, however not all SBERT models were successful. The two best performing models were the small SBERT-MiniLM (Map = 41.62) and SBERT-MPNet (Map = 44.76) while the original SBERT models (SBERT-base, SBERT-large and later SRoberta)(Reimers and Gurevych, 2019) showed lower performance. All these models have in common the use of encoder-only architecture for word representation and the Siamese network for sentence representation. The main differences lie in the used training data and more importantly the pretraining objective. BERT is based on masked language model pretraining and SRoBERTa is based on XLNET. Finally, BERT-MPNet is based on masked and permuted pretraining objective. If training data plays an important role in the quality of the training model, we believe that for SBERT, training objective with both XLNET or MPNET as well as token sequence length play a crucial role in providing higher performance.

An interesting finding is that our best combination models were obtained based on the small SBERT-MiniLM. This result shed light on the fact that bigger models do not necessarily mean better performance on one hand, and on the other hand that the fine-tuning strategies matter and can drastically impact the performance. If further investigations are certainly needed, using a small model such SBERT-MiniLM on larger data is greatly beneficial for both: training and inference time. We note that token sequence length for SBERT-MiniLM is 512 while SBERT-MPNet is only 128. Further investigations are certainly needed, by conducting for instance a controlled evaluation for patents and research papers as well as products of a length that fits the model or by developing new strategies that leverage truncated texts in their model.

Finally, the two generative LLM models (Llama3.1 and Gemma2) showed mitigated performance on zero-shot and few-shots learning. It it to note that the inference time of these models may increase with the number of shots. Our prompt strategy was to ask the LLM to provide a similarity score between -1 and 1 of two input sentences for zero-shot, and by presenting one positive sentence pair and one negative sentence pair for one-shot learning. We also conducted few shots learning but did not observe any significant improvement. Considering the complexity and size (billions of parameters) of these models, we believe that they are not a strategic choice for our task. We however, do not exclude the evaluation of smaller distilled versions of generative models in the future.

9 Conclusion

This work is the first attempt for modeling innovation by mapping research to the market. We conducted an extensive evaluation of a wide range of LLMs and showed under which conditions research paper information can be injected to improve the technology to market alignment. A thorough evaluation of sentence embedding models revealed that using SBERT with our fine-tuning strategies significantly improve the alignment performance. Furthermore, we showed that it is possible to map research with the market without using the technology information which is encouraging for building models able to recommend market products out of research papers. We hope that our findings and preliminary results will encourage new investigations in modeling innovation.

10 Limitations

If comparing several pretrained sentence embedding models on the technology to market alignment task has shown SBERT to be the most appropriate model for this task, it is worth noticing that the purpose or pretrained objective of several compared models was not semantic similarity. Which may indicate that this comparison is somehow unfair, and a better way to compare these models is to first fine-tune them on a semantic similarity training data before applying them to our task. Specter, a strong baseline should also be fine-tuned on our training dataset. We let this comparison for future work.

Also, our evaluation has been conducted on English only. If the results are promising, there is a need for evaluation on other languages. This is also part of our near future agenda. Finally, this work is a first attempt to model innovation. If we presented a methodology that can match research to innovation, the centroid representation of products does not allow a direct alignment with individual products. A further step is needed to extract each product from the market description. This post processing is part of our ongoing research.

11 Acknowledgements

This research is supported by the Research Center for Advanced Science and Technology (RCAST) of The University of Tokyo.

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, and colleagues. 2024. Gpt-4 technical report.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giridharan Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeffrey Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva, and Veselin Stoyanov. 2022. Efficient large scale language modeling with mixtures of experts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11699–11732, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, and colleagues. 2022. Palm: Scaling language modeling with pathways.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Team Gemma. 2024. Gemma 2: Improving open language models at a practical size.

Amir Hazem, Kazuyuki Motohashi, and Chen Zhu. 2024. From technology to market. bilingual corpus on the evaluation of technology opportunity discovery. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7510–7520, Torino, Italia. ELRA and ICCL.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Kenneth B. Kahn. 2018. Understanding innovation. *Business Horizons*, 61(3):453–460.

Athanasios P. Kalogeras and Christos Anagnostopoulos. 2012. Innovation modelling: Understanding the fundamentals of the transformation of research to innovation. *IFAC Proceedings Volumes*, 45(4):182–187. 1st IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Stephen Jay Kline and Nathan Rosenberg. 2009. An overview of innovation.

Yann Lecun. 1986. Learning processes in an asymmetric threshold network. In *Disordered systems and biological organization, Les Houches, France*, pages 233–240. Springer-Verlag.

Changyong Lee, Daeseong Jeon, Joon Mo Ahn, and Ohjin Kwon. 2020. Navigating a product landscape for technology opportunity analysis: A word2vec

approach using an integrated patent-product database. *Technovation*, 96-97:102140.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Matt Marx and Aaron Fuegi. 2020. Reliance on science: Worldwide front-page patent citations to scientific articles. *Strategic Management Journal*, 41(9):1572–1594.

Kazuyuki Motohashi and Chen Zhu. 2023. Identifying technology opportunity using dual-attention model and technology-market concordance matrix. *Technological Forecasting and Social Change*, 197:122916.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. OpenAI blog:12.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

F. Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323:533–536.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2022. Scirepeval: A multi-format benchmark for scientific document representations. In *Conference on Empirical Methods in Natural Language Processing*.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zheng, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022. Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model. *CoRR*, abs/2201.11990.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. One embedder, any task: Instruction-finetuned text embeddings.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony S. Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *ArXiv*, abs/2211.09085.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *ArXiv*, abs/2205.01068.