# Patent Search Using Triplet Networks Based Fine-Tuned SciBERT

Utku Umur ACIKALIN & Mucahid Kutlu

TOBB ETÜ University of Economics & Technology 2003

SIGIR 2022 Madrid

PATENTSEMTECH 2022

# Motivation

Increasing number of patents

Recent advancements in NLP

# PROPOSED APPROACH

- Representing patents with SciBERT embeddings
- Fine-Tuning via Triplet Networks
- Ranking Patents

# Patent Representation

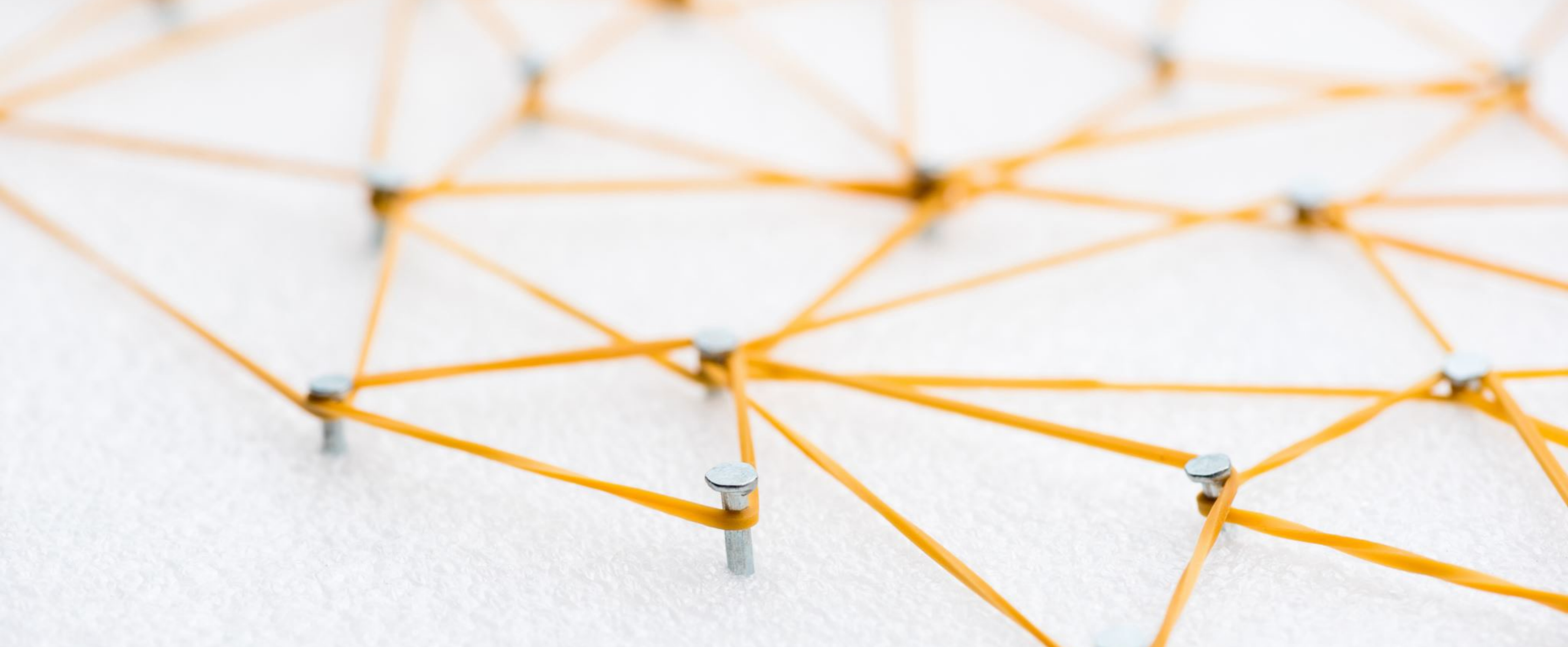
# Patent Representation

## How to tackle with technical language of patents?

- We use SciBERT which is pre-trained on scientific publications

## How to represent long patent documents using BERT models?

- **Create** separate embeddings for the description ($vd$) and claims ($vc$) part of each patent.
  - For long **descriptions**: Summarize via TextRank
  - For long **claims**: Truncate the parts that exceed the BERT limit
    - **Intuition**: the first claim of patents is generally the main innovative part of the patents while the other claims are less important ones.
- **Concatenate** the vectors for the description and claim parts
- **Normalize** them to have a unit norm
- **Give more weight** to the description parts than the claims



# Fine-Tuning via Triplet Networks

# Fine-Tuning via Triplet Networks

## We fine-tune SciBERT using Triplet Networks approach

- Allows us to derive fixed-size embeddings for each patent,
- Requires positive and negative samples for each patent to learn the semantic differences between relevant and not relevant patents.

## We construct 3 embeddings for each patent

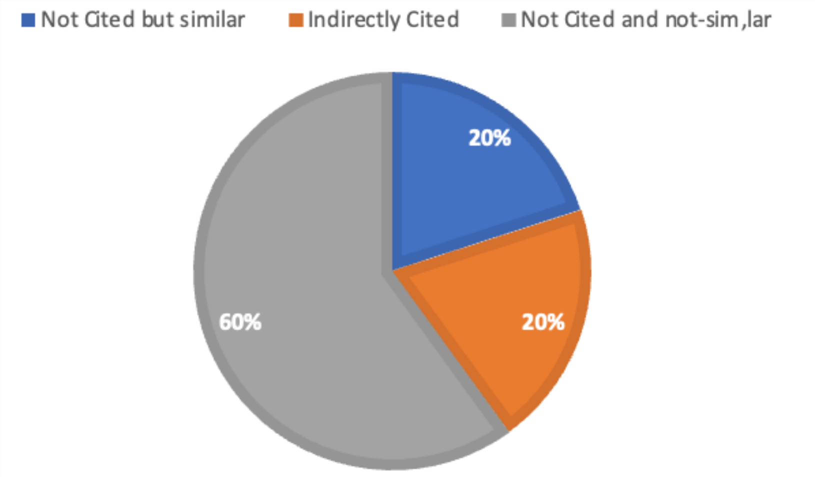
- an anchor patent (i.e., the patent itself) (a)
- a positive (i.e., relevant) patent (p),
- a negative (i.e., not relevant) patent (n).

## Triplet objective loss

- $max(CosineDistance(va,vp) - CosineDistance(va,vn) + \epsilon, 0)$

# Data Selection for Fine Tuning

- Positive Samples
  - The cited patents which have a similarity score of higher than 0.6 according to vectors provided by Google
- Negative Samples:
  - **Not Cited but similar ones**: the not-cited patents which are from the Cooperative Patent Classification (CPC) group of the anchor patent
  - **Indirectly Cited Ones**: the patents which are not cited by the anchor patent but cited by the patents that it cites.
  - **Not Cited and not similar ones**: Randomly selected from the patents which are not cited by the anchor patent and have a similarity score of less than 0.6 based on Google's vectors



# Experiments

## Dataset

- Randomly select 2 million patents granted after 1980.
- Among these patents, 1,817,504 of them have a title, abstract, description, and claims sections.
- From this sample, we randomly select 5,000 patents for testing, and others are used in training & search operations.
- We consider cited patents as relevant ones and not-cited ones as not-relevant.

## Training

- Train the model with four million examples (i.e., patent triplets)
- Use patents which have at least five backward and forward citations in total, as anchors in the training set.

# Results

| Ranking Method | Average Precision | Recall@100 | Recall@500 | Recall@1000 |
|---|---|---|---|---|
| Lucene with TF-IDF | 0.0548 | 0.2178 | 0.3642 | 0.4364 |
| Lucene with BM25 | 0.0469 | 0.1800 | 0.3083 | 0.3743 |
| Our Approach | **0.0675** | **0.2233** | **0.3934** | **0.4821** |

# Conclusion

- Proposed a novel method to represent patent documents by fine-tuning SciBERT with Triplet Network approach.
- Our proposed method outperforms baseline methods in our experiments.
- **What is next?**
  - Use other variants of BERT such as PatentBERT and other variants that have higher token limits.
  - Evaluate in various test collections
  - Compare against other baseline methods
  - Investigate which parts of patent documents are more important for the prior-art search task

# Thanks! Any questions?

This study was funded by the Scientific and Technological Research Council of Turkey (TUBITAK) ARDEB 1001 Grant No 119K986.