Universiteit van Amsterdam

MSc Artificial Intelligence
Master Thesis

Classifying Genes and Proteins as Drug Targets in Chemical Patents with Contextual Embeddings

by
Reinier Kruisbrink
11270721

July 29, 2022

48 EC
November - July

Supervisors:
A. Kakrana
W. Vlietstra

Examiner:
Prof Dr P. Groth

Second reader:
D. Cruz

ELSEVIER

Contents

Abstract

With the increasing amount of journal and patent publications in the biomedical domain, automatic information extracting approaches have become increasingly popular. Chemical patents are an important resource for chemical information. However, few information extraction systems have been evaluated on patent documents, due in part to their structural and linguistic complexity. In this thesis, we focus on extracting relevant genes and proteins by predicting whether they are direct or indirect drug-targets with contextual embeddings. We compared a machine learning baseline against embeddings based models. We created a silver standard text dataset by extracting contexts around gene and protein mentions in a set of chemical patents for we had relevancy labels. We improved upon the current state-of-the-art model and investigated several embedding approaches and classifiers.

Keywords: Drug targets, chemical patents, information extraction, contextual embeddings

Chapter 1

Introduction

1.1 Problem Statement

In commercial research and development projects, initial public disclosure of new chemical entities often takes place in patent applications [40]. Entities can be drug compounds, also called substances, which have an effect on a target; or drug targets, also called genes and proteins, which are affected by the compound. Out of a manually-curated set of 130 compound–target interaction pairs in patents only about 10% of the pairs could be found in publications in the scientific literature within one year of being reported in patents [41]. On average, it takes an additional 1 to 3 years for these chemical entities to be described in journal publications if at all [34]. Therefore, a large selection of these chemical entities is primarily available through patent documents [12]. Chemical patents describe advances in drug development such as information on new chemical compounds and the genes and proteins they target. This information can be used by e.g. pharmaceutical companies for competitive intelligence, which informs companies about the compounds and diseases their competitors are working on, thereby directing their research and investment efforts [7]. While the goal of scientific articles is to inform and explain, patents have the tendency to obfuscate any novelty that could benefit other companies [38]. Extracting the information about drug targets from patents is thus challenging given the obtuse language and mentions of other genes and proteins that are not directly related to the pathway affected by the drug. Therefore, identifying drug targets from patents currently requires manual annotation, which is a time consuming and expensive process.

In order to cope with the increasing amount of publications in the form of articles and patents in the biomedical and chemistry-related fields [23], automatic approaches for information retrieval (IR) and extraction (IE) have become increasingly popular. Applying natural language processing (NLP) to biomedical and chemistry-related publications gave rise to the field of biomedical/chemical text-mining [40]. Other approaches are applied to extract information from tables and images (e.g. chemical structures), which will not be discussed in this work.

A common NLP task and fundamental to chemical text-mining is named entity recognition (NER), also called chemical entity recognition (CER) in this context. For example, in order to extract relations between chemical entities, the entities themselves need to be extracted first. One of the difficulties with CER is the amount of synonyms that refer to the same entity in the form of brand names, various chemical notations and abbreviations. Normalization is frequently used to cope with these variations. Another task that heavily relies and even builds upon CER is the task at hand; relevancy scoring. Perfect entity recognition extracts all occurrences of chemical entities, even though many mentions may not be of much importance to the document (e.g. listings of similar entities or byproducts of reactions). All the entities that the authors have included are not of the same importance to every reader. Relevancy scoring aims to predict the importance of an entity to a document. Relevancy is a property that depends on signals derived both from their immediate vicinity, from its location in the patent, and domain knowledge.

There are many challenges, in general, in the automatic extraction of information from chemical patents. An issue, with this task specifically, is that there is no dataset available to train and evaluate deep learning models on. In particular, a textual dataset of patents required for the generation of contextualised embeddings does not exist as far as we know. There is a dataset of handcrafted features for machine learning approaches which is not suited directly for this task. Oftentimes research on information extraction in the biomedical domain is applied on scientific publications. For example, out of the currently eight available BioCreative open tasks, only one exists for patents [29]. Patents are textually much more complex due to the obfuscating language and therefore a less attractive field of research. There exists a dataset for the same task of relevancy scoring but for compounds, instead of the genes and proteins they target [44]. A comparable dataset was generated by weak supervision for the purposes of this thesis, which will be discussed in more detail later.

For this work, a relevant entity is a drug target and is defined as any gene or protein which is a direct or indirect target of at-least one drug or chemical compound mentioned in the patent. Such a relevant drug target is a document-level annotation. In order to predict relevancy labels, the necessary inputs are both the entity for which a label has to be predicted as well as the patent for which the label applies, as is shown in Figure 1.1. The prediction is based on features extracted for that entity in that patent, which can be handcrafted features or embeddings from extracted text. Subsequently, a model has to decide whether the provided entity is deemed relevant with respect to the provided contextual information. The provided context can, in theory, range from a single sentence to the whole document. Figure 1.2 shows one extracted sentence before it is embedded for *Progesterone receptor* for patent EP3275888A1, which is labeled as relevant.

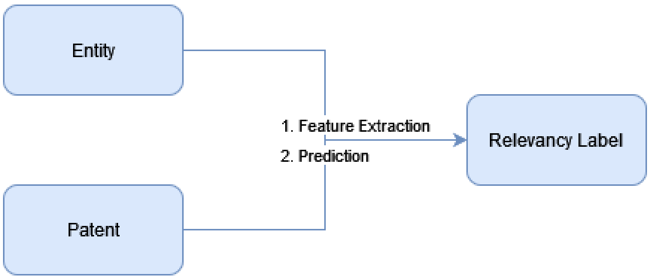
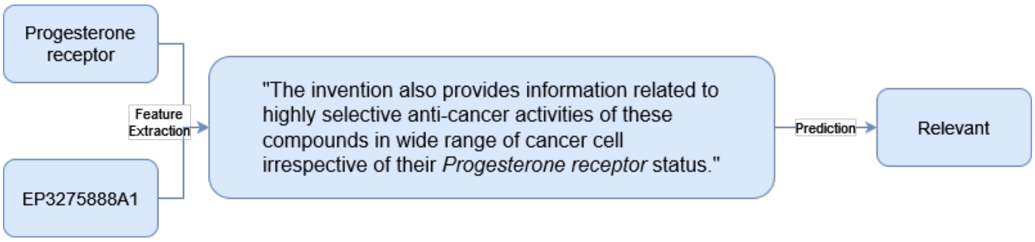


Figure 1.1: Visualisation of the task as input to output mapping.



"The invention also provides information related to highly selective anti-cancer activities of these compounds in wide range of cancer cell irrespective of their *Progesterone receptor* status."

Figure 1.2: An example result of feature extraction and subsequent prediction.

1.2 Research Question

The goal of this thesis is to explore and evaluate the use of contextual embeddings when classifying genes and proteins as drug targets, resulting in the following research question:
What added value do contextualised embeddings offer in the task of drug target relevancy scoring in patents?
The aim is to determine the added value by answering the following sub-questions:

- What is the baseline performance for both handcrafted features and contextualised embeddings, and how do they compare?
- What is the best performance that can be achieved for target-relevancy classifier for patents, and by what means?
- What are the reasons for the obtained model performances and what can be improved in future attempts of a similar task?

Contextual embeddings pre-trained on large text corpora, such as ELMo and BERT, move beyond global word representations like Word2Vec and achieve state-of-the-art performance on a wide range of natural language processing tasks [52]. Because contextual embeddings achieve state-of-the-art performance in almost all NLP tasks and the task at hand is very similar to text classification, it is hypothesised that the usage of embeddings will be able to improve on the currently deployed classifier.

1.3 Thesis structure

This thesis is structured in the following way: In chapter 2, preliminary knowledge on patents and the language models used for the generation of embeddings is reviewed. This includes transformers and their attention mechanism as well as classifiers built based on these embeddings. In chapter 3, the related work is highlighted by looking at similar tasks and research at Elsevier on this topic. In chapter 4, the problem of data availability is discussed and a solution is described by automatic generation of a dataset. Model architectures and representations of the data are illustrated for the experiments, as well as training and evaluation procedures of the final models. In chapter 5, we describe the experimental setup and present results. In chapter 6 we analyse the results and conclude, and propose directions for future research.

Chapter 2

Theoretical Background

This chapter will describe the preliminary knowledge on the topics to tackle the challenges at hand. This includes a short description of chemical patents, the fundamental NLP techniques used for embeddings and the reasoning behind them.

2.1 Patents

> *A patent is a type of intellectual property that gives its owner the legal right to exclude others from making, using, or selling an invention for a limited period of time in exchange for publishing an enabling disclosure of the invention [1].*

All patents are structured in a similar fashion. Title and abstract come first, followed by the claims and description. There is, however, a significant difference between patents and scientific publications. The focus lies on demonstrable ownership, as opposed to scientific publications, where the focus lies on the sharing and spreading of knowledge [38]. The key difference is therefore that patents are not meant as a source of information (even though they may very well be!). This is precisely the reason for the obfuscating language used in many patents. For example, patent claims are often made of complex, long sentences with multiple clauses and dependencies to other claims [21].

Besides deliberately hiding information, there are many more *unintended* challenging aspects of extracting information from chemical patents. Firstly, patents need not go through the same peer review process as journal publications. Patents depend on whatever requirements patent laws and patent offices in a specific country put on the claims put forward by the patentee. But no country requires another researcher to scrutinize the claims made by a patent. Secondly, in the chemical domain, the text is more densely filled with chemical formulas and various chemical notations interrupting the flow of running text [20]. Aside from the text, important information is captured in images of chemical reactions or in tables with numerical results. Training a model to aggregate information from these different modalities poses a challenge for IE in chemical patents. Thirdly, patents come from patent offices all around the world with the consequence that patents may be entirely or partly written in different languages [28]. For example, one single patent document may include certain sections in French while some other sections are in English. Machine translation of these sections is imperfect which induces a loss/alteration of information and the possibility of cascading errors in downstream models.

2.2 LightGBM

The baseline model, against which the classifiers using contextual embeddings are compared, is a fairly recent and very effective machine learning method. LightGBM, short for Light Gradient Boosting Machine, is open source gradient boosting framework for machine learning originally developed by Microsoft [39]. Decision trees is one of the foundations, on which LightGBM is built [16]. In a decision tree, each internal node represents a split on a data feature, and each branch represents the outcome of such a split. Until a leaf node is reached where a decision is made. The paths from root to leaf represent classification rules. The model is a way to display an algorithm that only contains conditional control statements and is therefore extremely simple, and yet profoundly effective in many cases. In combination with decision trees, gradient boosting [9] is used. It is a machine learning technique that gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted decision trees (GBDT). Formally, the predictor F is an ensemble of various weak learners h_i in Equation 2.1. The best predictor $\hat{F}$ is the result of supervised learning on the output variable y and a vector of input variables x, and the minimization of some loss function $L(y, F(x))$, which is depicted in Equation 2.2.

$$F(x) = \sum_i (\gamma_i h_i(x)) + c \tag{2.1}$$

$$\hat{F} = argmin_y \ \mathbb{E}_{x,y}[L(y, F(x))] \tag{2.2}$$

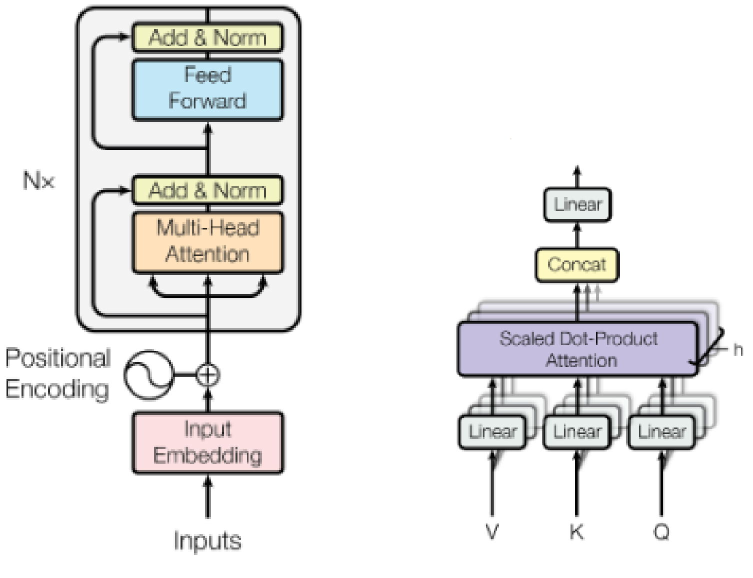
The weak learners, h_i, are called weak learners because they are some constraints that can be imposed on the construction of decision trees. Constraints imposed include number of trees, tree depth, number of nodes or number of leaves among others. They can be constructed in a greedy manner, choosing the best split points based on minimizing the loss or maximizing information gain. An important insight into creating ensembles is allowing trees to be greedily created from random subsamples of the training dataset. This same benefit can be used to reduce the correlation between the trees in the sequence in gradient boosting models. This variation of boosting is called stochastic gradient boosting. This essentially specializes each weak learner in a subset of the data and improves the performance of the collection of weak learners.

Both the number of instances and the number of features have significantly increased in recent years with the explosive growth of data. This poses a challenge for GBDT algorithms as they need to, for every feature, scan all the data instances to estimate the information gain of all the possible split points. This especially affects the trade-off between efficiency and accuracy. LightGBM, introduces two novel techniques to scale GBDTs with big data. Instead of random subsamples from the data, Gradient-based One-Side Sampling (GOSS) is proposed. The researches behind LightGBM noticed that data instances with different gradients play different roles in the computation of information gain. They keep those training instances with large gradients, where large is defined as among the top percentiles or by a pre-defined threshold, which will contribute more to the information gain. And they randomly drop the ones with small gradients, thus effectively reducing the required amount of training instances. As for the amount of features, they propose Exclusive Feature Bundling (EFB). This is a near-lossless method to effectively reduce the amount of features. In a sparse feature space many features are nearly exclusive, implying they rarely take nonzero values simultaneously. A perfect example of exclusive features are one-hot encoded features, where only a single entry has a non-zero value. EFB bundles these features, reducing dimensionality to improve efficiency while maintaining a high level of accuracy.

2.3 Embeddings

Even though models like LightGBM perform well on specific and simpler NLP tasks, they are not suited for more complex tasks such as summarization and text generation. They are classifiers, and the open issue remains whether they understand language or just learned the task. Most natural language tasks have two aspects in common, namely, syntax and semantics. Learning these two fundamental parts of natural language before starting text classification or text generation is transfer learning. The idea is to generalize to a language model which allows for better performance in downstream tasks. Implementations of general language modelling range from co-occurrence counts in semantic vector space models, which are quite inefficient (due to sparsity), to dense deep learning representations.

This works considers Transformers for the creation of the contextual embeddings. One of, if not the most, common Transformer language models nowadays is `BERT` [43], which achieved state-of-the-art performance in various language-based tasks. They instantiated a model consisting of 12 Transformer blocks, each with 12-headed attention, resulting in 110M parameters. This model was trained on two unsupervised tasks, namely Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). MLM enables the model to create representations by predicting masked tokens based on surrounding tokens. NSP allows for understanding relationships between two sentences, which is not directly captured token prediction. Their model was pretrained on both the BooksCorpus (800M words) [36] and the English Wikipedia (2.5B words) excluding headers, tables and lists (which are quite common in patents).



(a) (b)

Figure 2.1: Transformer Architecture (a), Attention Mechanism (b). From Vaswani et al. [42]

Figure 2.1b shows the architecture of a single head attention mechanism in the Transformer model. Attention is a function that maps a query and a set of key-value pairs to an alignment score between the query and the key. This essentially enables the model to pay attention to the important parts of the input. The alignment can be computed between two tokens as is shown in Equation 2.3, where Q is the query, K is the key, V is the value and d_k is the dimensionality of the key used for scaling. The query and the key are the inputs on which the alignment has to be computed. This attention score is used as a weight to alter the value of the token embedding, such that it includes alignments with other tokens. In multi-head attention, a single attention layer is divided in multiple smaller section to enable focusing on different relationships and thus allow for different alignments between inputs (think of agent-verb, adverb-verb, and adjective-noun alignments). Self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation, and the self-alignment thereof, of the same sequence. Cross attention mixes two different embedding sequences, such as a sentence in English and one in French in a machine translation task. This is also possible for different modalities, like a text embedding and an image embedding to form text captions for images.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.3}$$

Figure 2.1a shows the architecture of the Transformer encoder model consisting of N stacked layers. This encoder model is responsible for generating contextual embeddings. As opposed to RNNs, Transformers contain no recurrence and therefore lose the positional information of the input sequence. In order to still capture some information about the relative or absolute position of the tokens in the sequence, positional encodings with the same dimension as the embeddings are inserted at the bottom of the model. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, fully connected feed-forward network. Residual connections and layer normalization are added after each sub-layer. Each sub-layer outputs embeddings of the same size to allow the residuals and positional encodings to be added directly to the embeddings. Due to the fact that there is no recurrence in this model, the maximum sequence length is pre-defined. Shorter input sequences are padded while longer input sequences are truncated. A Transformer model can be initialised for longer and longer sequences if needed. However, attention scales quadratically ($O(n^2)$) with the input sequence length, which results in a high complexity for extremely long sequences. Attention variations have been proposed to mitigate the high memory usage and computation times. One of those variations is Longformer [48].

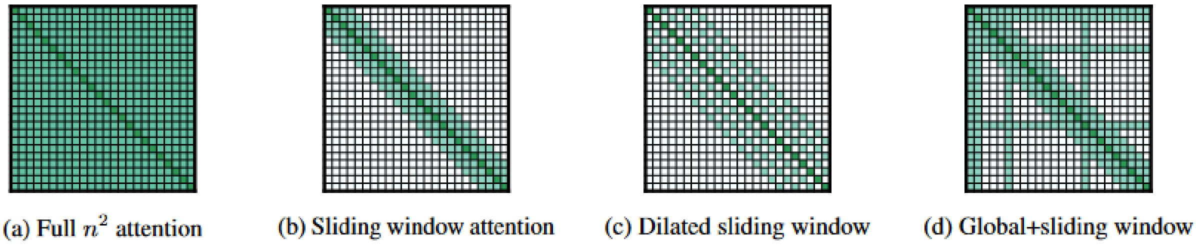

(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

Figure 2.2: Longformer Attentions. From Beltagy et al. [48]

Figure 2.2a shows the full attention matrix, whereas b, c, and d show estimations, which can be combined to improve performance on longer sequences while also reducing the complexity to a linear scale ($O(n + w + s)$, where w is the window size, and s is the selection of input tokens for global attention). The window is capable of capturing local alignments. Stacking multiple layers of such windowed attention results in a large receptive field, where top layers have access to all input locations and have the capacity to build representations that incorporate information across the entire input, similar to CNNs. To further increase the receptive field without increasing computation, the sliding window can be dilated. The global attention on only a few input tokens captures some of the larger alignments which has a negligible impact on the complexity because $n \gg s$.

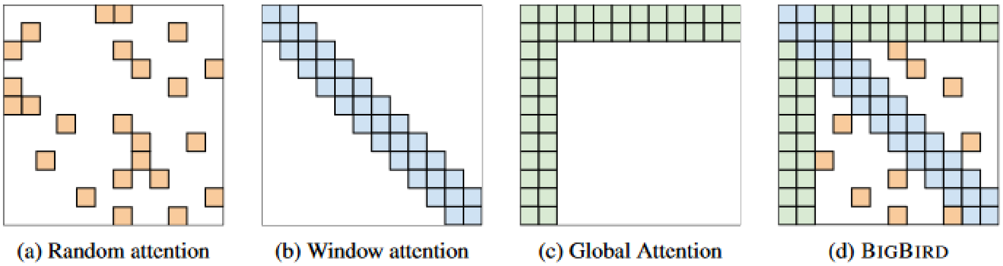


Figure 2.3: BigBird Attentions. From Zaheer et al. [54]

Another variation is `BigBird`[54] and shown in Figure 2.3. The full attention matrix is reduced to an almost linearly dependant complexity ($O(n+w+r)$, where w is the window size, and r is random attention), and allows sequences up to 4096 (8 times the maximum sequence length of `BERT`).

2.4 Classifiers

After contextualised embeddings of e.g. sentences have been generated, a fully connected feed forward layer can be trained to perform an NLP tasks such as text classification. If the text classification is on a paragraph containing multiple sentences, and since each sentence was embedded, the paragraph is now represented as a sequence of sentence embeddings. For this task, RNNs such as LSTMs [5] and even CNNs [27] with 1-dimensional convolutions can be employed for more complex pattern recognition and classification of the input embeddings as sequences.

Chapter 3

Related work

This chapter will explain the related works for relevancy scoring as well as tasks integral to relevancy scoring such as chemical entity recognition. An extensive review of the field of chemical text-mining up to 2017 was created by Krallinger et al. [40]. Since then various deep learning approaches have been explored to improve upon the rule- and dictionary-based or machine learning based models that were frequently employed. Several community-wide efforts for evaluating information extraction and text-mining developments in the biomedical and chemical domain have been hosted, for example the Biocreative challenges [13]. Shared tasks include text-mining for gene entity recognition [19], chemical compound and drug name recognition in articles [30] and in patents [29][55], drug chemical-protein interactions [58], and many more. Other challenges that are hosted every few years are BioNLP-OST [24][25], which focuses more on the biomedical domain, and ChEMU [50][56] for more chemistry-related tasks.

3.1 Natural Language Processing on Patents

In general, NLP has been applied to various kinds of documents, from technical documentation to World Wide Web, to automatically extract information. There are many differences between highly structured CSV files and highly unstructured texts such as running text. In between lie semi-structured texts such as XML and structured documents such as patents. Consequently, information extraction systems must be tailored to the needs of the task. A tool capable of identifying the core of a patent invention automatically to check the relevance of a patent with minimal efforts is very useful, but also very different from the automatic summarization of a novel. Cascini et al. [11] applied text mining methods in 2004 to patents and technical documentation such as software requirements texts to show that these techniques are able to effectively support the human comprehension of these classes of technical documentation. More modern approaches do extract the information automatically instead of supporting people in finding information.

Aras et al. [26] introduce three text mining tools specifically designed for patent texts. In many technical fields, key information is provided in the form of figures and units of measurement. However, when these data appear in full text, they are almost certainly lost for search and retrieval purposes. They propose the numeric property extraction. They also show the challenges of keyword extraction, such as subjectivity between reader and author, neologisms and heterogeneous text (tables, math, chemical formulas). These problems require deeper analysis of the content, to understand patent texts better and improve searching specific aspects or entities in the texts. Finally, they propose a tool to automatically segment the description of a patent into sections and identify the parts which constitute to a patent more easily.

Most of the work on text mining in the biomedical and chemical domain, i.e. extracting chemical entities and their relationships, has been applied to scientific journals. Since 2010, the US Patent and Trademark Office (USPTO) publicly offered its patents online. The increased availability of patents created an opportunity for researchers to design and improve specialised algorithms for patents to extract the knowledge contained therein. Besides the availability of patents, two community-wide efforts for evaluating information extraction in biomedical and chemical patents have been hosted. Firstly, the Biocreative V CHEMDNER patent task [29], which presents the results of the first named entity recognition assignment carried out to detect mentions of chemical compounds and genes/proteins in patent titles and abstracts. Secondly, ChEMU lab (Cheminformatics Elsevier Melbourne University) provided two reference resolution distinct tasks in chemical patents [55]. The first one aims to identify chemical reactions conditions specified in the description of another reaction. The second one aims to identify the reference relationships between expressions in chemical reaction descriptions.

3.2 Fundamental tasks

When extracting information from textual data in the chemical doman, named entities are essential. E.g. for drug-protein interactions, drugs and proteins need to be recognised as entities before a relation can be found between the two. Similarly, in order to classify genes and proteins as drug targets, the genes and proteins need to be recognised before they can be classified.

As a consequence, chemical entity recognition (CER) is fundamental to most other tasks. Several approaches have been explored to perform entity tagging. One of the first and obvious approaches is to use dictionary and rule-based methods to find entities [10]. These methods strongly rely on domain-knowledge and hand-crafted rules. These dictionaries and rules can be extended and updated to capture specific needs and the most recent changes in the field. Over the past two decades, a range of different statistical machine learning approaches and supervised learning algorithms have been tested for NER problems [15], including but not limited to random forests (RFs) [31], support vector machines (SVMs) [35], hidden Markov models (HMMs) [6], and conditional random fields (CRFs) [17][32][33]. CRFs are a class of statistical modelling methods often applied in pattern recognition and machine learning. Whereas many other classifiers predicts a label for a single sample without considering *neighbouring* samples, a CRF can take context into account. These models use hand-crafted features such as N-gram occurrence frequencies and word shapes. Akhondi et al. [37] improved on Leaman et al. [32] by combining dictionary and statistical methods. Modern deep learning approaches rely less on domain-experts who craft features and rules but more on annotated datasets, which are expensive and time consuming to create. Since there is still some discussion about what constitutes an entity in the chemical domain and if the goal of the task changes or a slightly different definition is used for annotation, the dataset may become obsolete. The combination of statistical and deep learning results in state-of-the-art CER, with CRF-BiLSTMs and contextualised word embeddings [47] with an F1 score of 0.86.

Another task is entity normalization, where the goal is to link entities mentioned in a document to standard database identifiers. After the recognition of an entity it is difficult to compare and cross reference the contents of documents with each other due to many synonyms (acronyms, various chemical notations, and brand names). Gene normalization is such a fundamental task, and therefore also hosted in the first three editions of BioCreative [13][18][22].

3.3 Relevancy Scoring

Chemical entity recognition exists to solely extract chemical entities from patents, which means they focus on all mentions and mentions only. Less attention has been given to the relevancy of an entity in a patent. A relevant entity plays a major role within a patent. Filtering between relevant and irrelevant entities in chemical patent texts was first suggested by Akhondi et al. [44]. An annotated dataset was created by multiple annotators with compound mentions and their corresponding relevancy scores. Statistical features of the entity in the patent where the entity was found were extracted. A classifier was trained on a linear combination of the following features: compound frequency, compound section, compound length and surrounding characters. The relevancy was chosen based on a cutoff which was determined by performance on the validation set. Instead of hand-crafted features, many modern NLP models use deep learning to create contextual embeddings, which may give a much richer description of the sentence and thus allow for better performance in relevancy classification. The compound relevancy classification system from Akhonid et al. [44] had an F1-score of 0.86.

Relevancy scoring is part of a larger project at Elsevier to process biomedical and chemical documents and extract information from them. In this project, relevancy scoring is preceded by chemical named entity recognition, and normalization (based on an internal gene and protein taxonomy).

Chapter 4

Method

This chapter will explain all the data used and how it was generated as well as the models used in the experiments in the next section. For this study, we will test the current state-of-the-art models for text embeddings. The performance achieved with embeddings is compared against the performance achieved with several handcrafted features, as well as to the combination of embeddings and handcrafted features. Both machine- and deep-learning models will be trained and evaluated as classifiers for these feature sets.

4.1 Data

Patent	Entity	Label
EP327588A1	Progesterone receptor	1
EP2200613B1	IKCa1	1
EP2200613B1	insulin	0
EP2268285B1	helicase	0
EP1585548B1	insulin	0

Table 4.1: Example triples, each triple constitutes one instance. Multiple entities can be mentioned per document, and likewise, an entity can be mentioned in multiple documents.

In order to predict relevancy scores, the necessary inputs are the entity and the patent for which a label has to be predicted as well as the features on which the prediction is based, as was shown in the Introduction in Figure 1.1. An internal set of around 240k hand-labeled triples had been annotated, consisting of a patent number, an entity identifier, and a relevancy label (Table 4.1). All patents were selected from American (US), European (EP) and Worldwide (WO) patent offices, on the basis that they were originally written in English. Due to document availability, a portion (Table 4.2) of all the patents in these triples was used generate a silver standard text dataset. The train/test splits are random non-overlapping subsets (70%, 30%) from the total set. There are patents and entities which occur in both these sets, but any combination of patent and entity is unique to each set as to minimize the information flow from train to test set. The irrelevant entities heavily outnumber the relevant ones but the ratio remains the same among both the train and test set. All entities are genes or proteins, all relevant entities are either direct or indirect targets. The reasons behind this are simply the guidelines for the manual excerption. Handcrafted features are used for the baseline and compared to contextual embeddings. The next two sections will describe what these features entail and how they were obtained.

	Patents	Entities	Relevant	Irrelevant	Total Instances (100%)
Train	831	5350	1082 (5.9%)	17364 (94.1%)	18446
Test	805	3296	450 (5.7%)	7456 (94.3%)	7906
Total	835	6385	1532 (5.8%)	24820 (94.2%)	26352

Table 4.2: Unique patents and unique entities, as well as the amount of positive and negative labels. The full set contains 26352 binary relevancy scores.

4.1.1 Handcrafted Dataset

There are a total of 27 features per document-entity pair, of which most are described below. There exist 18 distinctive features because some are normalised versions or combinations of other features. E.g., term frequency is included as well as the normalised term frequency in the set of 27.

One of the most commonly used features in text classification is term frequency–inverse document frequency (tf-idf), and represent represents a measure of how much information the word provides, if it is common or rare across all documents [3]. Besides document wide tf-idf values, section specific tf-idf values are also extracted from the dataset. The assumption behind this is that an entity that occurs multiple times in the abstract is more likely to be relevant than multiple occurrences in the description. Other word frequency features that are extracted are `MechCooccurrence`, which counts how many times the entity is mentioned in the same sentence with a *mechanism of action* keyword. Mechanism of action (MoA) is the biochemical interaction through which a compound produces its pharmacological effect (e.g. inhibition). The genes and proteins, such as enzymes and receptors, to which a compound binds or affects is frequently mentioned with a mechanism of action. Similarly, `InvCooccurrence` counts how many times the entity is mentioned in the same sentence with an *invention* keyword (e.g. invention, formula, compound). Invention keywords denote importance with respect to the invention of a patent and thus may also denote importance of the relevancy of an entity. Also included is the character count of the entity's longest available synonym in the document. For compound relevancy scoring, the entity length turned out to be one of the most important features because oftentimes substances have its full chemical (IUPAC) notation somewhere mentioned in the patent if they are of any relevance and those notations are in general long. The final features that are extracted are Booleans that signify whether the first occurrence was in the title or abstract, or whether it was first mentioned in the claims or description sections, `abs_title_first_occ` and `claim_des_first_occ`. All these features are derived from the text of the document, even though not all of the information in the patent comes in the form of text. Images and tables include some of the most important entities as well. Therefore, a feature whether the entity was also mentioned inside a table was included.

The list of most distinctive features is tf, df, idf, tf-idf, tf per section, tf-idf per section, mechanism of action keyword co-occurrences, invention keyword co-occurrences, entity length, first title/abstract mention, first claims/description mention, table mention.

4.1.2 Text Dataset

Dataset Generation

A dataset consisting of text is required to generate embeddings. A silver standard dataset is generated by weak supervision for the generation of contextual embeddings. Weak supervision is the method of labeling large amount of training data in a supervised setting by using noisy, limited or imprecise sources. The actual textual training instances inherit their label from the triples, which can be executed automatically and reduces the need for hand-labeled data at the loss of data quality. Nonetheless, strong predictive models can be trained as a result of this semi-supervised labeling method.

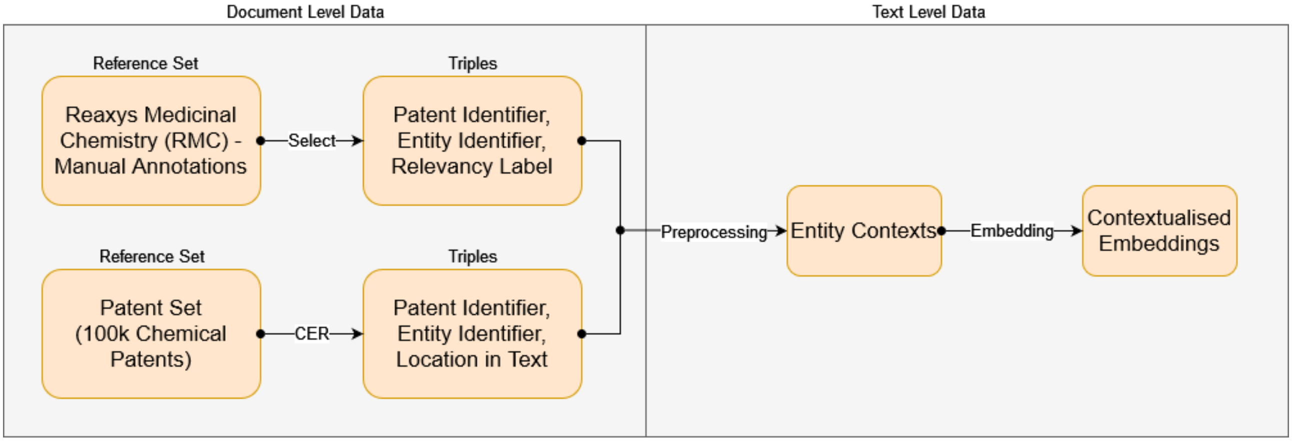

Document Level Data
Text Level Data
Reference Set
Reaxys Medicinal Chemistry (RMC) - Manual Annotations
Select
Triples
Patent Identifier, Entity Identifier, Relevancy Label
Reference Set
Patent Set (100k Chemical Patents)
CER
Triples
Patent Identifier, Entity Identifier, Location in Text
Preprocessing
Entity Contexts
Embedding
Contextualised Embeddings

Figure 4.1: An overview of the generation of the textual dataset and embeddings.

All patents are passed through Elsevier's CER and normalization tools to obtain normalised entity identifiers for all different mentions as well as their character location in the text. In order to extract the relevant text snippets around the entities for which we have labels, a window on both sides of the entity is extracted by a pre-defined character offset. Because the text is in XML format the text snippet has to be cleaned to become pure text. Sentence segmentation is applied to the resulting text and the outermost sentences are dropped because they are incomplete sentences. The applied sentence tokenizer uses the Punkt algorithm [14], trained on chemistry text. Very short sentences (less than 5 tokens) are also dropped because they contain little information and are probably artifacts of the sentence segmentation. Masking of non-focus entities[1] by their group name (chemical or gene or protein) is applied to reduce the amount of out-of-vocabulary tokens coming from these chemical entities and make sure other mentions map to a single token.

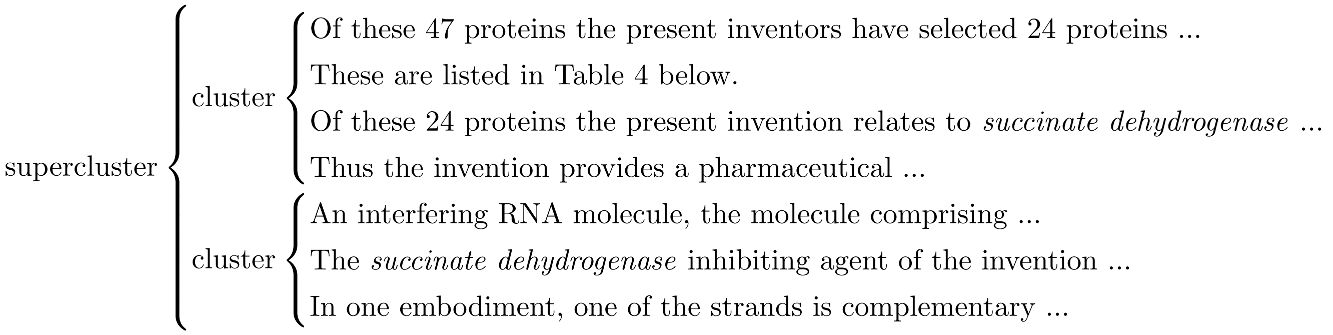
At this point, the data consists of a few sentences before and after the sentence in which the entity was mentioned. Together these sentences form, what we call, a *cluster* (similar to a short paragraph). So, there exists per patent, per entity mention, a cluster of sentences. Due to the possibility of multiple mentions near each other, there may be an overlap between clusters. If an entity is mentioned in two consecutive sentences, both these sentences (and their neighbouring sentences) are extracted twice, resulting in duplicate sentences. To remove the duplicate sentences, a dataset hyperparameter is defined which allows overlapping clusters to be added together into a larger cluster, depending on the amount of overlap. The overlap hyperparameter is set to 0, meaning that clusters are added if they have directly neighbouring

[1] Non-focus entities are other genes and proteins that were tagged by LeadMine (One of Elsevier's internal CER systems), as well as chemicals that were tagged by ChemDataExtractor2.0 [57].

Algorithm 1 Pseudocode of the preprocessing step in Figure 4.1

```
Require: PatentSet $P$
  CharOffset ← 2000
  MinSentsPerClust ← 3
  MaxSentsPerClust ← 10
  Overlap ← 0
  for Patent $p \in P$ do
      TaggedEntities ← CER($p$)
      for Entity, Location ∈ TaggedEntities do
          Context ← ExtractContext(Entity, Location, CharOffset)
          Context ← CleanString(Context)
          Sentences ← SentenceSegmentation(Context)
          Cluster ← Combine(Sentences, MinSentsPerClust, MaxSentsPerClust, Overlap)
          Cluster ← Normalize(Cluster, Entity)          ▷ Normalize focus entity
          Cluster ← Mask(Cluster, Entity)               ▷ Mask non-focus entities
      end for
  end for
```

sentence and also if there is overlap. Setting it to 1 would mean that the clusters are only combined if they share at least one sentence. Two other hyperparameters are introduced to constrain the minimum and maximum sizes of clusters in terms of sentences. The minimum and maximum of sentences per cluster are set to 3 and 10 respectively. If more than 10 sentences exist per cluster, the first 10, from the beginning of the document, are kept. The cutoff of 10 sentences per cluster is based on the distributions shown in Figure 4.3 such that at least about 80% of the data remains. For the purpose of this work, these hyperparameters stay the same but it may be interesting to see how the models perform on variants of the dataset.



Of these 47 proteins the present inventors have selected 24 proteins ...

These are listed in Table 4 below.

Of these 24 proteins the present invention relates to *succinate dehydrogenase* ...

Thus the invention provides a pharmaceutical ...

An interfering RNA molecule, the molecule comprising ...

The *succinate dehydrogenase* inhibiting agent of the invention ...

In one embodiment, one of the strands is complementary ...

Figure 4.2: Dataset levels visualisation.

The point where weak supervision comes in, is that the labels are defined per document, and we just extracted data on the level of sentences. Document level labels are assigned to data on the level of a few sentences. The problem is that not every sentence/cluster contains a signal towards the relevance of the entity and thus introducing a label alignment problem. The label is aligned to the entire document. The closest representation there is of the entire document in this dataset is the combination of all clusters, which we call a *supercluster*. Three levels of the dataset are defined: sentence, cluster, and supercluster, each one corresponding to the level at which the label can align with the text. The example in Figure 4.2 visualises the relation between the various levels of the dataset for the entity *succinate dehydrogenase*. Each line of text is a sentence. This example tells that *succinate dehydrogenase* has three mentions in this specific patent, of which the first two were close to each other and had overlap and were thus combined, whereas the last mention forms its own cluster. The supercluster is simply the combined clusters as a single piece of text, even though, in reality, the components are not consecutive pieces of text.

The supercluster is truncated after 5 clusters because a significant amount of the superclusters contain less than 5 clusters and some go up to as much as hundred clusters which are assumed to not contribute much additional value. If more than 5 clusters exist for an entity in a document, the first 5, from the beginning of the document, are kept. The cutoff of 5 clusters per supercluster is based on the distributions shown in Figure 4.3 such that at least about 80% of the data remains.

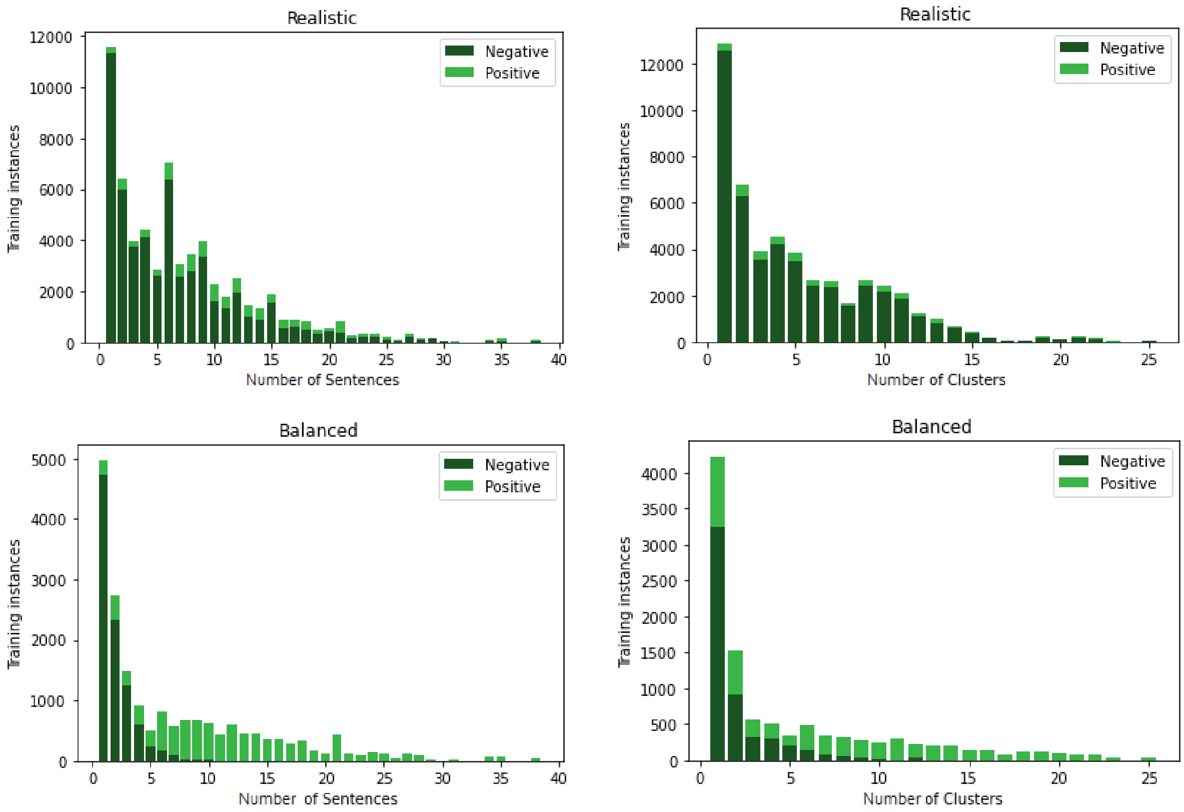


Figure 4.3: Extracted sentences (w. mention) and clusters per document-entity-label triple for both a realistically distributed set as well as a balanced set. (Note that for the balanced set, more mass lies in the tail as relevant entities tend to have more mentions and thus more extracted sentences and clusters)

Label Alignment Problem

Table 4.3 shows that out of the original 26532 training instance triples there are now substantially more depending on the level of the dataset. There are 21551 superclusters, which are document representations and thus should align with the amount of original triples but since not every entity was recognised or able to be extracted with context, less remain. However, multiple clusters are extracted per entity per document, and even more sentences. In total there are 47553 clusters, consisting of 183127 sentences of which 64798 sentences contain the corresponding entity. About a third of the total amount of sentences contain an entity, because one sentence on both sides are included as context per cluster and thus about one in three sentences has an entity mention.

	Number
Triples	26532
Supercluster	21551
Cluster	47553
Sentence (total)	183127
Sentence (w. mention)	64798

Table 4.3: Total amount of extracted text snippets per data level.

The task is to predict a relevancy label per document-entity pair. Now that there are, e.g., multiple extracted sentences for such a triple, the original labels do no longer align one-on-one with the text instances that were extracted. The problem is that it is possible to predict based on any one of them. E.g., to determine which of the sentences provided as an example in Figure 4.2 should be used to predict the relevancy label for *succinate dehydrogenase* for that document, several approaches are tested by choosing various contexts during the prediction. One of the simplest approaches is to predict based on only the first mention, assuming that the first mention is one of the most important and informative. Another approach combines multiple sentences in such a way that more information is included and allows the true label to align with the data. In the next chapter, these approaches and their results will be explored to resolve the problem of choosing what contextual information to base the prediction on and thus how to align the original label to the extracted text.

Generating Embeddings

`BERT` is trained on a lot of text which includes many domains, allowing it to generalize very well on text sequences in general domains. However, the task at hand deals with extremely domain specific texts and thus would benefit from a specialised model, analogous to how people would first learn to understand general texts before they would attempt to read a specialised document such as a chemical patent. The first transfer learning step is to train a Transformer language model like `BERT` on general text understanding. Subsequently, another step of transfer learning can be performed by specializing the language model on task or domain specific texts [49]. Task adaptation would be to pre-train on texts similar to the ones which are inputs for the task so it becomes familiar with them. In tasks like question answering there is such a clear structure which would improve the model to understand questions and corresponding answers. For relevancy scoring, which is essentially text classification, there is no such a structure. Yet, the domain has substantially different language than the general domain. For this reason there exists a variation like `BioBERT` [51], a specialised continuation of the original model. `BioBERT` was initialised from the general `BERT` model and was subsequently domain adapted to the biomedical domain by pre-training on PubMed Abstracts (4.5B words) and PMC Full-text articles (13.5B words).

	Seq. Length (avg, max)	Model	Embedding dim.
Sentence	34, 465	`BioBERT`	768
Cluster	132, 762	`BigBird`	1024
Supercluster	291, 2645	`BigBird`	1024

Table 4.4: Amount of training examples per level

Table 4.4 shows sequence length statistics (in terms of tokens) for each level of the dataset. `BERT` has a maximum sequence length of 512 which means it is not able to fully embed both the cluster level and the supercluster level. The reason for this is the quadratic dependency on the attention computation, which `BigBird` [54] resolves. Unfortunately, no domain adapted model exists for Transformers that allow for longer sequences. The sentence level dataset is embedded by `BioBERT` as a 768 dimensional vector while both the cluster and supercluster level datasets are embedded by `BigBird` as 1024 dimensional vectors. The embeddings were generated by an implementation in Huggingface's Transformer library [46]. In particular, `monologg/biobert_v1.1_pubmed` and `google/bigbird-roberta-large` were the model cards that were used.

4.2 Models

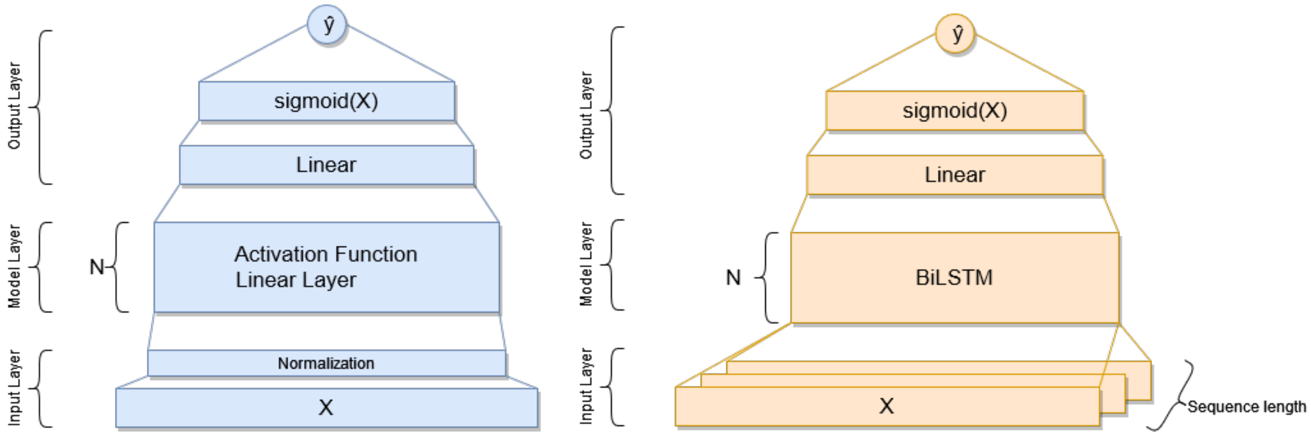
In order to determine and evaluate the performance of various models, the baseline is a simple model with the handcrafted features described in 4.1.1. The Python LightGBM API was used to train on all the handcrafted features. Additionally, a multi-layer perceptron (MLP, see Figure 4.4a) was trained on the same features. Where LightGBM uses weak decision trees to split on input features, the MLP uses those same input features as numerical values. A conditional split in a decision tree does not depend on the scale a feature, but solely on the difference between those values. Because some of the input features are raw word counts and others are Boolean values there is a large discrepancy between their numerical values as input for the MLP. E.g. a raw word count of 95 would entirely dominate a Boolean value of 1. The input normalization methods that were implemented and included as a hyperparameter for the training of this second handcrafted feature model are shown in Equation 4.1 and refers to the normalization in the input layer of Figure 4.4a. Internal model normalization is achieved through ReLU activation functions.

$$\begin{aligned}
x_s &= \frac{x - min(x)}{max(x) - min(x)} && \text{(minmax scaling)} \\
x_s &= \frac{x - \mu}{\sigma} && \text{(standard scaling)} \\
x_s &= \frac{x}{||x||} && \text{(unit scaling)} \\
x_s &= tanh(x) && \text{(tanh scaling)} \\
x_s &= log(x + 1) && \text{(log scaling w. Laplace smoothing}^2\text{)}
\end{aligned} \tag{4.1}$$

When it comes to fine-tuning a model on the embeddings, the straightforward approach is to add an MLP on top of the pre-trained embeddings and fine-tune this classifier on the relevancy score prediction task (as is shown in Figure 4.4a, with the embeddings as input and without the input normalization layer). Currently the embeddings are the representation of the entire input text, e.g. because the clusters are embedded directly. Besides fine-tuning a simple MLP classifier, it is also possible to add more complex models as classifiers which will be able to learn more patterns inside the embeddings.

Similar to how a sentence can be represented as the sequence of single word embeddings, so too, can a cluster be represented as the sequence of sentence embeddings. Each sentence is embedded as a 768 dimensional vector and there can be up to 10 sentences per cluster,

[2] Because any logarithm is undefined at 0 and the data includes Boolean zeroes, all values are smoothed by adding 1. All 1-valued features would be mapped to 0, adding 1 alleviates this problem. All other values increase by 1 as well, such that the ordinality persists. This is also known as additive smoothing (https://en.wikipedia.org/wiki/Additive_smoothing).



(a) (b)

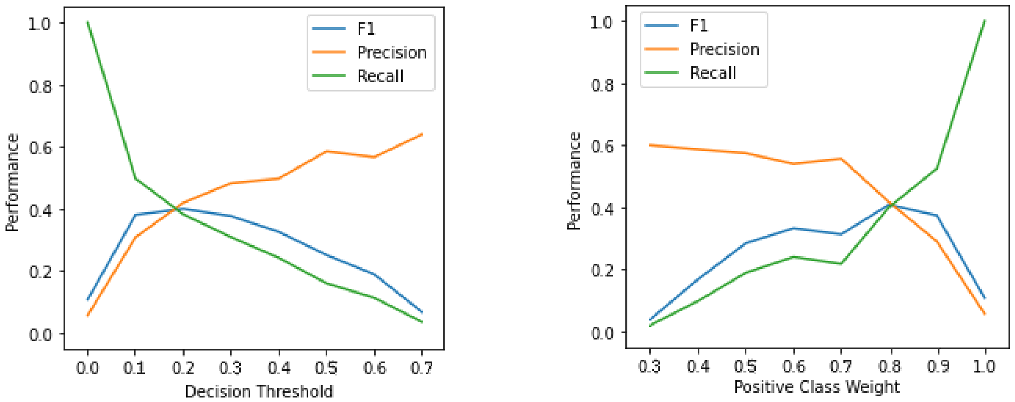
Figure 4.4: The MLP architecture (a) and the sequence processing model (b).

resulting in 10 × 768 shaped inputs. This means we have an additional way of representing clusters in terms of lower level embeddings. What this also means is that it is possible to use RNNs or CNNs to classify these representations because of their sequential nature. Figure 4.4b shows the architecture that was implemented to process the new input representations, which we call *sent2clust* based on the fact that they combine sentences into clusters. One way to characterise the difference between the direct cluster and sent2clust representations is that the direct embedding combined its text components *before* embedding, whereas the sent2clust representation was combined *after* the embedding of its components.

The final model that was implemented is a hybrid model, because it combines both handcrafted features and the contextual embeddings, effectively incorporating document wide and local context features in one single model. The hybrid model architecture is as simple as the input and model layers (like in Figure 4.4) without their individual output layers. Instead a shared classification layer is added as a single linear layer with sigmoid activation to predict a relevancy score $\hat{y}$. The relevancy score $\hat{y}$ is turned into a relevancy label prediction based on a threshold, which is optimised during the hyperparameter tuning.

4.3 Training

One of the issues with predicting relevant genes and proteins is the severe class imbalance. In our dataset, only 6% of the entities are marked as relevant. Imbalanced data can lead to complications when training binary classifiers due to the majority class dominating the learning process. Methods like class weighing [4] and over-/undersampling exist to neutralize such imbalances. Class weights directly modify the loss function by giving more (or less) penalty to the classes with more (or less) weight. When it comes to oversampling, new samples are either synthetic samples or duplicates of existing samples. Advanced synthetic oversampling, such as SMOTE [8], may generate flawed minority class training samples which will confound the learning process. Synthetic oversampling will not be explored here because embeddings for both classes looked very similar after preliminary tests by UMAP clustering and would presumably not result in useful synthetic samples. Duplicating existing samples is very similar to class weighing as they both result in more impact of the duplicated/weighted samples, which are already seen during training. In the next chapter, experiments are performed with both class weighing and undersampling of the minority class. Figure 4.5b shows an example of the effect of weighing the minority class more. In our implementation, increasing the minority class weight automatically reduces the majority class weight. The negative class weight is defined as $w_{negative} = 1 - w_{positive}$, while the positive class weight is included in hyperparameter search to optimize the performance. In addition to class weighing, Figure 4.5a shows an example of the effect of the decision threshold for classification, which is also included in the hyperparameter search. Beside the possibility to optimize for F1 scores, it is also possible to configure a model differently if you want to optimize more for precision or recall.



(a)

(b)

Figure 4.5: Performances for (a) classification thresholds and (b) positive class weight. The threshold and the positive class weight were kept at 0.5 while testing the other value. The reported performances are from an MLP with the handcrafted features described in Section 4.1.1.

Hyperparameter search is performed by branch and bound algorithm design paradigm [2] implemented in Optuna [45]. As opposed to exhaustive grid search, branch and bound uses lower and upper bound performance metrics as a heuristic. The set of candidate solutions is thought of as a tree, in which branches are subsets of the hyperparameter search space. The algorithm explores branches of this tree by checking against upper and lower estimated bounds on the optimal solution, and is discarded if it cannot produce a better solution than the best one found so far. The algorithm depends on efficient estimation of the lower and upper bounds of branches of the search space. If no bounds are available, the algorithm degenerates to an exhaustive search. The optimization criterion is the validation set F1 score, which performed better than validation set loss.

4.4 Evaluation

To connect with the broader research community and subsequent attempts at target relevancy scoring, the models will be evaluated using commonly used performance metrics such as precision-, recall-, and F1-scores. These metrics are well suited for imbalanced classification tasks such as this one, as opposed to a metric like accuracy. The evaluations will be done based on a held out portion of the generated dataset; these include manually annotated genes and proteins that were considered as relevant (direct or indirect targets) for the chemical patents, and automatically extracted sentences. A validation set is used for model selection and hyperparameter tuning. The test set results are run and evaluated on 10 seeds as to mitigate different model initialisations and to yield a better indication of real world performance.

Chapter 5

Experiments & Results

This chapter describes the experiments and their results. The consecutive experiments can be seen as versions, where consecutive experiments build upon the best performing models and their assumptions of previous experiments.

5.1 Baseline

For the machine learning baseline, a LightGBM model was trained on 27 handcrafted features and included in all result plots to compare against the models trained on embeddings. The baseline was trained with an equal class weight for the minority and majority class and is shown in the Figure 5.2 and Figure 5.3. By introducing class weights and an additional feature (whether the entity was also mentioned in a table) we were able to improve the baseline from 0.43 F1 to 0.52, which will be shown as baseline in all other experiments except for the first. LightGBM is a deterministically generated model and thus has a single performance instead of a performance estimate averaged over several runs. Figure 5.1 show all feature importances and Table 5.1 correlations for some features. There is something to note about the third most important feature: entity length (in characters). This was also the case for compound relevancy scoring as chemical notations can get quite long, yet this is not necessarily the case for genes and proteins. A reason behind this could be the fact that novel entities are not shortened or referred to as acronyms yet because they are only recently discovered.

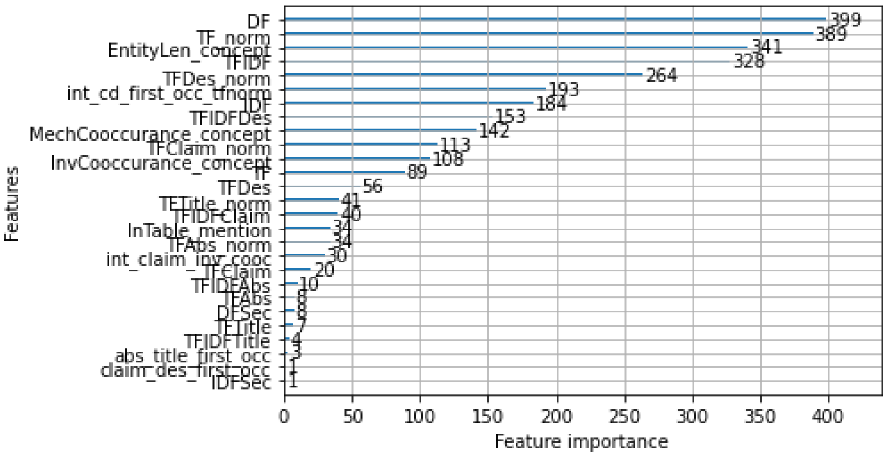


Figure 5.1: Handcrafted feature importances.

Average term- and document frequencies show an expected pattern for both classes. Relevant entities tend to be mentioned more often in a document than irrelevant entities. And relevant entities tend to be mentioned in fewer documents because they are presumably more recently explored/discovered genes and proteins. Even though only few entities are mentioned in tables at all, relevant entities are mentioned more than 10 times as often in a table. The last two features show that these co-occurrences strongly correlate with the relevancy label.

Class	TF	DF	Entity Length	Table Mention	MechCooc	InvCooc
0	5	141	8	0.008	0.83	0.66
1	21	107	10	0.1	6.5	6.4

Table 5.1: Mean handcrafted feature values for both classes.

5.2 Label Alignment Experiment

In this experiment several classifiers are trained on various embeddings. The goal of the experiments in this section is to discover which level of the dataset contains the most predictive power. First, to make sure there is one single representation of each level per label, only the first mention is used to base the prediction on (as was described in Section 4.1.2). For each model the resulting input shapes are shown in Table 5.2. This means that each document-entity pair is represented as either a single embedded sentence, or a single embedded cluster, or an embedded supercluster. The (very basic) assumption is that the first mention is one of the most important and informative of all the mentions in a document. This also means that each models (`sent`, `clust`, `supclust`) contains more information in terms of amount of text than the previous, respectively.

Level	Shape
sent	1 × 768
clust	1 × 1024
supclust	1 × 1024

Table 5.2: The input data shapes for this experiment

One single sentence embedding achieves almost the same F1 score as the supercluster which contains much more information, which reveals that probably most of the important information is included in the sentence with the mention itself. However, the dataset on the sentence level is embedded by `BioBERT` as opposed to `BigBird` for the cluster and supercluster level. Preferably, the quality of sentence embeddings should be combined with the amount of context included in a supercluster.

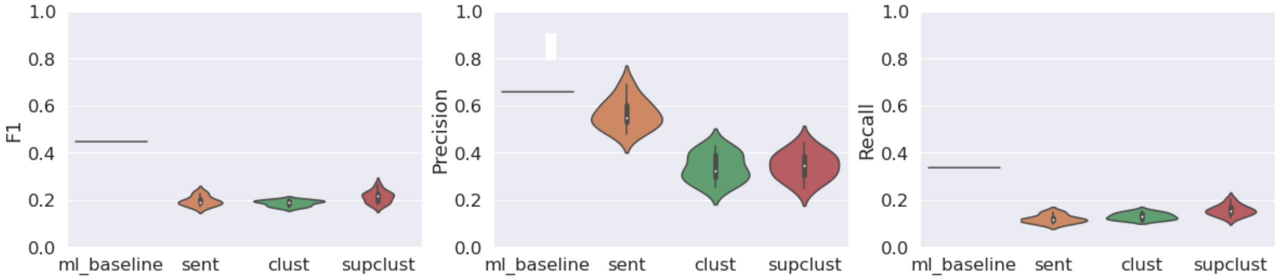


Figure 5.2: Simple Label Alignment Experiment

In the second experiment, the majority class was undersampled to such an extent that the class labels are now balanced. This balanced set is used to train the same three models for each level of the dataset and the results are shown in Figure 5.3. The results do not show any improvement on the models trained on the full dataset. Instead of removing data by undersampling the majority class, the next experiment considers other ratios of positive-to-negative class labels without removing any data by using class weights.

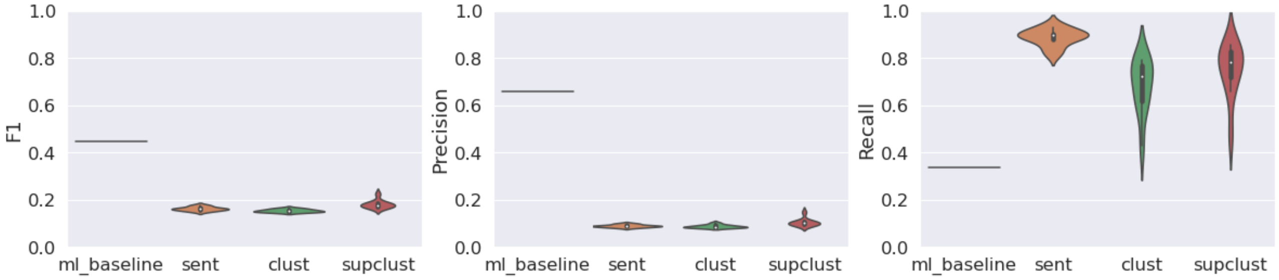


Figure 5.3: Balanced Label Alignment Experiment

The same three models are trained again, but with optimal class weights as to increase the impact of the minority class. Additionally, a lower decision threshold is employed to neutralize the tendency of the deep learning models to produce lower scores because the 0 class label is the majority class. Even the baseline LightGBM model performance increases significantly, from 0.42 F1 to 0.52. The sentence and supercluster models improve from 0.21 F1 to 0.25, while the cluster level only increases from 0.19 F1 to 0.21.

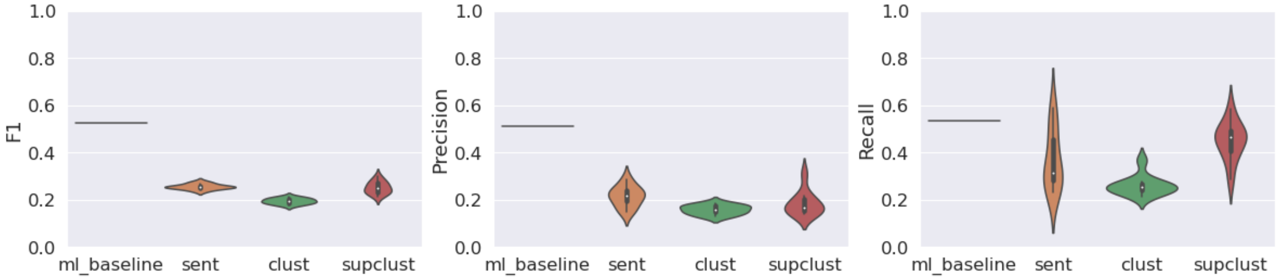


Figure 5.4: Optimised Label Alignment Performances

5.3 Aggregating Embeddings Experiment

The outcomes of the experiments in the previous section are that the sentence embeddings perform as well as the supercluster, despite using a different approach. The goal of the experiment in this section is to aggregate the lower level embeddings to combine the quality of the lower level embeddings with the amount of information included in superclusters by increasing the amount of sentences included in the new models. For the first model, sentence embeddings are combined to form a cluster after each individual sentence was embedded and hence the name (sent2clust). Each cluster consists of 3 to 10 sentences, resulting in an input shape of 1 × 10 × 768, which is padded to 10 if the cluster consists of fewer than 10 sentences. This model was explained in Section 4.2 and depicted in Figure 4.4b. The model uses a bidirectional LSTM to compute a cluster representation from the sequence of sentences and thus has more contextual information around the sentence with the mention. Because sent2clust predicts based on a cluster representation, only the first cluster of the document is used. This model improves on the previous model on cluster level from 0.19 F1 to 0.31, but also on the sentence and supercluster level from 0.25 F1 to 0.31.

Level	Shape
sent2clust	$1 \times 10 \times 768$
supsent	$1 \times 6 \times 768$

Table 5.3: The input data shapes for this experiment

For the second model, only sentences with an entity mention are used. Instead of sentences around one single mention, multiple sentences with a mention are used. However, these sentences are not necessarily consecutive sentences and thus should not be processed with an RNN like the other model. Up to 6 sentences are concatenated and passed through an single layer MLP classifier. The same model was tested with up to 3 and 9 sentences yet performed arguably the best without adding unnecessary complexity after 6 sentences. Table 5.4 shows that the F1 score does not improve much after adding more sentences. This model combines the effective sentence embeddings with a larger representation by including multiple mentions, like a supercluster, and hence the name supsent for short. This model achieves an F1 score of 0.395, the highest so far.

#Sentences	F1	Precision	Recall	#Parameters
3	0.362	0.282	0.545	295k
6	0.395	0.366	0.434	590k
9	0.398	0.361	0.451	884k

Table 5.4: Performance for different amount of sentences included in the supsent model.

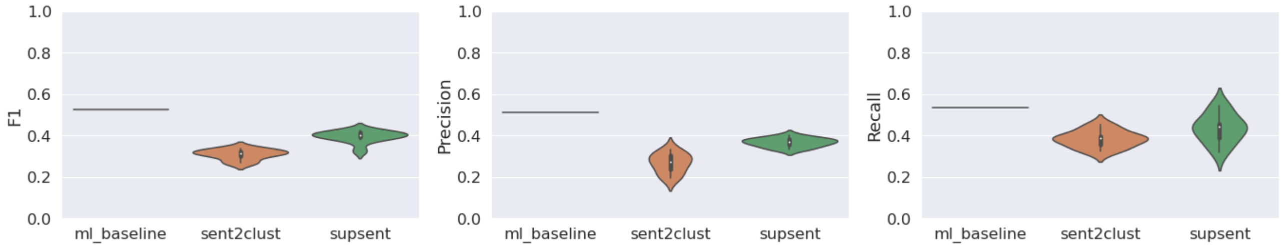


Figure 5.5: Aggregated Embeddings Experiment

5.4 Hybrid Data Experiment

The experiments in the previous section reveal that increasing the amount of information the model has improves model performance but only up to a point. However, more information should allow the model to make better predictions. The handcrafted features can be included in the model to allow it have document- and corpus-wide information on the document-entity pair. In order to sideload the handcrafted features and determine if it would improve upon the aggregated sentence embedding models from the last section, a single layer MLP is trained on all distinctive handcrafted features. The hybrid model combines the best performing embedding model with these handcrafted features by fine-tuning two single layer MLPs with a combined classification layer. Figure 5.6 shows that the handcrafted features perform almost as well as the best embedding model, while all models so far are still outperformed by the improved LightGBM model using class weights and one additional feature. The hybrid model achieves an higher F1 score than the original baseline, while it is just slightly lower than the improved baseline.

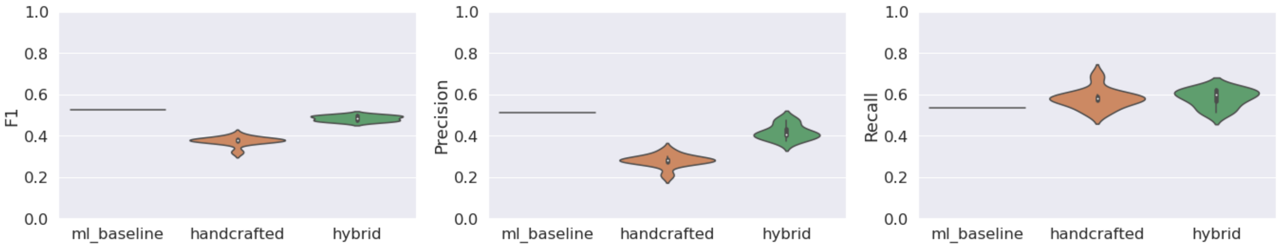


Figure 5.6: Hybrid Models: handcrafted features and embeddings

5.5 Qualitative Analysis

In this section, sentences before embedding will be analysed. The goal is to investigate where and why the model makes mistakes. To this end, predictions from the best performing embedding model, which is supsent, are taken from two categories based on the difference between prediction and true label. First, 25 false negatives, i.e. relevant instances that were predicted the most irrelevant by the model. And second, 25 false positives, i.e. irrelevant instances that were predicted as the most relevant by the model.

When looking at high confidence true positives, the first and foremost observation to point out is that all sentences run smoothly as text because they contain only a few proper nouns and are properly segmented and grammatically correct. When looking at the false negatives, however, many sentences are practically illegible due to being incomplete and/or filled with proper nouns, such as entity mentions and procedures. The false positives mostly have well formed sentences again that are very similar to those of the true positives, which is probably why the model (falsely) predicts them as positives.

Table 5.5 show the counts for a few noteworthy reasons that may cause trouble for the embedding and the subsequent classification. The following 8 aspects of input sentences will be counted for both categories:

1. List, the text contains a listing of multiple other entities.
2. Ref, the text is a citation or reference to another document or table/image and thus does not itself contain the information it tries to convey.
3. Recipe, the text is a recipe that specifies conditions under which a reaction takes place or how an entity was produced.
4. Single Sentence, only one sentence with a mention was extracted and thus contains little information.
5. Seg. Error, a sentence in the text is erroneously segmented and is thus incomplete.
6. Mask. Error, most other entities (such as in a listing) are not properly masked and thus result in many out-of-vocabulary tokens.
7. MoA, the text has mechanism of action keyword co-occurrences.

8. Inv., the text has invention keyword co-occurrences.

Below a few examples from the false negatives will be highlighted to show the basis on which the first 6 aspects were chosen. All of these are single sentences in order to make the examples less cluttered. The last two aspects were chosen based on the fact that they were chosen by domain experts and included as handcrafted features that represent positive signal (also described in section 4.1.1).

> 1) `Muller-Enoch, D., E. Seidl, and H. Thomas, [Chemical (Chemical) as a substrate for Catechol O-methyltransferase (peptide) (author's transl)].`

Example 1 contains a citation to another document which is accompanied by a segmentation error. Two non-focus entities are properly masked by the their group name *Chemical.* It is obvious that any model would have a hard time extracting valuable information regarding the relevancy from the embedding of such a sentence. Yet, *Catechol O-methyltransferase (peptide)* is a relevant drug target for the document.

> 2) `Cells and compound are incubated for 1h at 37°C, 5%Chemical before stimulation with 5ul/well recombinant Bone morphogenetic protein 6 (R&D Systems #Chemical) at a final concentration of 100ng/ml.`

Example 2 describes a process which has to do with the focus entity *Bone morphogenetic protein 6*, while two other non-focus entities are again masked properly. Even though this sentence is complete and syntactically sound, it does not seem to contain a lot of information towards the relevancy of this entity w.r.t. the entire document. And since this sentence is the only information to base the prediction on, it may not look relevant, yet it is labeled as such.

> 3) `Controls - GeneOrProtein Chemical; TRPV2: tranilast; TRPV3: Chemical; TRPV4: Chemical; TRPV5: Chemical; GeneOrProtein, Chemical; GeneOrProtein: Chemical; TRPC3: Pyr3; TRPC4: ML 204; TRPC5: ML 204; TRPC6: ML 204; TRPC7: GeneOrProtein, TRPM2: Chemical; TRPM3: GeneOrProtein acid; TRPM4: Chemical; TRPM5: Chemical; Transient receptor potential cation channel subfamily M member 8: Chemical.'`

Example 3 is a sentence which should contain information about the entity *Transient receptor potential cation channel subfamily M member 8* and its relevancy. This example is a listing of many other compounds and genes and proteins, of which only some are recognised and masked by their group name *GeneOrProtein* and *Chemical.* Even if all were masked properly, it still seems impossible for a model to extract from this example that the focus entity is, in fact, a relevant entity.

	List	Ref	Recipe	Single Sentence	Seg. Error	Mask Error	MoA	Inv.
False Neg.	6 (24%)	5 (20%)	6 (24%)	7 (28%)	4 (16%)	5 (20%)	6 (24%)	2 (8%)
False Pos.	0 (0%)	1 (4%)	1 (4%)	2 (8%)	0 (0%)	0 (0%)	15 (60%)	7 (28%)

Table 5.5: Counts of sentence aspects that impact the embedding/classification (out of 25 in total).

Frequently, an input is/contains a listing, a reference, or a recipe, but not all three at once. Out of the total of 25 inputs, 6 are/contain listings, 5 are/contain references, 6 are/contain recipes, and 6 are just a single sentence. There is some overlap between these impediments, with the result that 80% (20/25) of the inspected inputs have flaws that interfere in the syntax and semantics of the text and, consequently, the embedding and classification. Besides the quality of the text, instances for which less contextual information exist, i.e., where only a single sentence was extracted, are less likely to be predicted as relevant. The false positives have significantly fewer flawed sentences and more extracted sentences per instance. They also have much more co-occurrences with keywords that are expected to co-occur with relevant entities (see Table 5.1).

5.6 Results Summary

Table 5.6 shows a single overview with all model performances.

	F1	Precision	Recall
LightGBM (original)	0.423	**0.649**	0.32
LightGBM (improved)	**0.523**	0.513	0.533
sent (98.6K)	0.249 (±0.017)	0.227 (±0.083)	0.382 (±0.163)
clust (131K)	0.194 (±0.012)	0.158 (±0.019)	0.261 (±0.04)
supclust (131K)	0.251 (±0.025)	0.184 (±0.049)	0.45 (±0.077)
sent2clust (1.6M)	0.309 (±0.023)	0.266 (±0.044)	0.383 (±0.038)
supsent (590K)	0.395 (±0.027)	0.366 (±0.02)	0.434 (±0.066)
handcrafted (2.9K)	0.374 (±0.02)	0.279 (±0.028)	0.580 (±0.044)
hybrid (625K)	0.484 (±0.012)	0.418 (±0.034)	**0.582** (±0.04)

Table 5.6: All model performances

Observation 1. The best overall performance was achieved by the LightGBM model using handcrafted features and class weights.

Observation 2. The models using sentence embeddings perform better than the model using cluster and better or as well as the model using supercluster embeddings.

Observation 3. Including neighbouring sentences as more context improves the performance.

Observation 4. There seems to be a point where including more mentions and their context does not further improve the model performance.

Observation 5. Looking at a small subset of misclassifications, 80% of them have at least one impediment in the text before they are embedded or classified.

Chapter 6

Discussion

With the vast amounts of compound- and target-related information being published in patents, text mining approaches are used for automatic extraction and to save the expenses and time of manual excerption. Chemical patents contain a myriad of entity mentions, while only a fraction of these are related to the invention disclosed in the patent. It is mostly this fraction that consists of novel compounds, targets and interactions and is thus of interest to researchers and pharmaceutical companies. The goal of this thesis was to explore and evaluate the use of contextual embeddings when classifying genes and proteins as drug targets in chemical patents. We set out to determine the added value of contextualised embeddings in the task of drug target relevancy scoring in patents. We set a baseline and compared it against classifiers on various representations formed by contextual embeddings by generating a silver standard dataset. We hypothesised to outperform the handcrafted features based baseline because contextual embeddings have done so in many text classification tasks.

There are several reasons why the embedding models achieved the performance they did. The results from the first experiment can be explained by two facts. First, class weighing does not remove data whereas balancing the dataset does. It is likely that balancing the class labels by removing instances from the majority class is too strict and removes too much data. Positive class weighing retains all instances and performs better. Second, the dataset on the sentence level is embedded by domain adapted `BioBERT` as opposed to `BigBird` for the cluster and supercluster level, which is presumably the cause for better performance considering the amount of contextual information the model receives (Observation 2). Including more context in the second experiment does improve the model up to a certain point (Observation 3 and 4). One of the main reasons for this is the fact that there are few instances for which a lot of context is even extracted, such that allowing it to include more context does not affect most instances. Another reason for this could be that later sentences contain the same or similar information as previous ones and therefore do not add any new information.

6.1 Conclusion

First of all, we were able to outperform the baseline by using a deep learning hybrid model with both handcrafted features and contextual embeddings. However, we also outperformed this baseline by using a LightGBM model that uses the handcrafted features by adding class weights to mitigate the severe class imbalance. The added value of the contextual embeddings was rendered obsolete by the improvement to the baseline (Observation 1). Our conclusion is that the generated dataset was the bottleneck for the performance. The text in this dataset was not good enough to generate embeddings, which are good enough to outperform the LightGBM model using the specified features. The classifiers performed nearly optimally with what they were given as input.

There are several reasons why the LightGBM performance surpasses that of the embeddings based approaches described in this thesis. The first one is that many extracted sentences, contained in the automatically generated dataset, hinder the generation of informative embeddings (from Observation 5). Contrarily, the extraction of most handcrafted features is not influenced by the syntax and semantics of the written text. E.g. co-occurrence counts rely solely on the proximity of two tokens. Additionally, generating embeddings relies heavily on the language model and the training data it has seen, whereas most of the handcrafted features do not rely on the domain of the text at all. Even though some of the embeddings are, in fact, domain adapted, it is not perfectly transferable from biomedical texts to chemical patents. Some of the handcrafted features, such as the co-occurrence counts of domain specific keywords chosen by domain experts, are also forms of domain adaptation. The main difference between these two versions of domain adaptation is that the embeddings are noisy signals and the handcrafted features are controlled signals. The embeddings contain that same co-occurrence (and much more) while the handcrafted feature simply is that co-occurrence (without any noise). All of the reasons above lead to more noise in the embeddings. And finally, it may be that the current amount of data, and in particular the amount of positive instances, is not enough to find as much of a pattern in the noisy data as in the handcrafted data.

There are also several limitations that may have impacted the results of this work. First, the goal was to determine the value of contextual embeddings in relevancy scoring even though no dataset existed for this purpose. So, a significant portion of the project was spent on the generation of the dataset. Because the dataset is a prerequisite for classifiers to be trained and evaluated, the generation may have been too rushed to start obtaining results as that was the objective. Errors and imperfections have cascaded from the automatic generation of the dataset to the predictions and results from the classifiers. Another limitation is that the train and test set split is currently on document-entity pairs. This results in every document-entity pair in the test set to be unseen, while it may have seen the document or entity already in the training set (just not the combination). In order to minimise the information flow between the training and test set, this split could be implemented on documents only. By this approach, the test set would be more realistic because newly obtained data in the production pipeline is also from new documents.

6.2 Contributions

1. We outperformed the baseline model and thereby achieved state-of-the-art performance in drug-target relevancy scoring.

2. We provide a dataset for future work on drug-target relevancy scoring.

3. We created a method for the generation of the dataset with several hyperparameters to control the specifications of the dataset.

4. We are the first to use contextual embeddings to predict genes and proteins as drug-targets and thereby explored the caveats and improvements as a starting point for future research.

6.3 Future Research

Since this was the first application of contextual embeddings to drug target relevancy scoring in patents to our knowledge, we have several suggestions for subsequent attempts to address the limitations and also discuss possible improvements and future research directions.

First, the generated dataset contained about 26k instances for which text was extracted from 835 patents that we had access to. There are, however, labels for about 240k instances for which we did not have all the patents. If the models were trained on this larger set, it could improve both the handcrafted features based approaches and the embeddings based approaches. To better guarantee the quality instead of increasing the quantity, perhaps the automatic generation of the dataset should be a standalone project. These two approaches still generate a silver standard dataset. For the comparable task of compound relevancy scoring a gold standard dataset was annotated and resulted in an F1 score of 0.86 by Akhondi et al. [44] (with handcrafted features) and an F1 score of 0.98 by another UvA-Elsevier thesis by Jonas Klass (with contextual embeddings). There is not much reason to believe the task is significantly harder for drug targets than for compounds. So a gold standard dataset would presumably allow for improvements for drug target relevancy scoring with contextual embeddings as well.

Considering the efficiency of the handcrafted features, both on their own and in combination with embeddings, another direction is to introduce more handcrafted features. As opposed to the direct embedding of sentences (or multiple sentences), text can also be represented by embedding individual words [53] and training classifiers to run over the sequence of embeddings as is done by other research at Elsevier on target relevancy scoring on scientific journal publications. This same research also includes more modalities than just text. Instead of exclusively considering textual data for this task, other modalities such as tables and figures can be included because they often contain valuable information on the relevant entities of a document.

Acknowledgements

I am most grateful to Elsevier for the opportunity of this research internship and allowing me to work with such a great team. In particular, I would like to thank my supervisors Atul Kakrana and Wytze Vlietstra for their guidance and effort during this project, as well as Camilo, Dicle, Umesh, Kinga and Abhinav for their insights and time.

I would also like to thank the UvA and all its employees for all the learning they facilitated for me during my time here, and special thanks to Paul Groth for his supervision.

Bibliography

[1] United States Court of Appeals (6th Circuit). *Herman v. Youngstown Car Mfg. Co.* 1911.

[2] Eugene L Lawler and David E Wood. “Branch-and-bound methods: A survey”. In: *Operations research* 14.4 (1966), pp. 699–719.

[3] Thorsten Joachims. *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization.* Tech. rep. Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[4] Claire Cardie and Nicholas Howe. “Improving minority class prediction using case-specific feature weights”. In: (1997).

[5] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[6] Daniel M Bikel et al. “Nymble: a high-performance learning name-finder”. In: *arXiv preprint cmp-lg/9803003* (1998).

[7] Pierrette Bergeron and Christine A Hiller. “Competitive intelligence”. In: *Annual Review of Information Science and Technology (Arist)* 36 (2002), pp. 353–90.

[8] Nitesh V Chawla et al. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[9] Jerome H Friedman. “Stochastic gradient boosting”. In: *Computational statistics & data analysis* 38.4 (2002), pp. 367–378.

[10] Meenakshi Narayanaswamy, KE Ravikumar, and K Vijay-Shanker. “A biological named entity recognizer”. In: *Biocomputing 2003.* World Scientific, 2002, pp. 427–438.

[11] Gaetano Cascini, Alessandro Fantechi, and Emilio Spinicci. “Natural language processing of patents and technical documentation”. In: *International Workshop on Document Analysis Systems.* Springer. 2004, pp. 508–520.

[12] Mervyn Bregonje. “Patents: A unique source for scientific technical information in chemistry related industry?” In: *World Patent Information* 27.4 (2005), pp. 309–315.

[13] Lynette Hirschman et al. *Overview of BioCreAtIvE: critical assessment of information extraction for biology.* 2005.

[14] Tibor Kiss and Jan Strunk. “Unsupervised multilingual sentence boundary detection”. In: *Computational linguistics* 32.4 (2006), pp. 485–525.

[15] David Nadeau and Satoshi Sekine. “A survey of named entity recognition and classification”. In: *Lingvisticae Investigationes* 30.1 (2007), pp. 3–26.

[16] Carl Kingsford and Steven L Salzberg. “What are decision trees?” In: *Nature biotechnology* 26.9 (2008), pp. 1011–1013.

[17] Roman Klinger et al. “Detection of IUPAC and IUPAC-like chemical names”. In: *Bioinformatics* 24.13 (2008), pp. i268–i276.

[18] Alexander A Morgan et al. "Overview of BioCreative II gene normalization". In: *Genome biology* 9.2 (2008), pp. 1–19.

[19] Larry Smith et al. "Overview of BioCreative II gene mention recognition". In: *Genome biology* 9.2 (2008), pp. 1–19.

[20] Bernd Müller et al. "Abstracts versus full texts and patents: a quantitative analysis of biomedical entities". In: *Information Retrieval Facility Conference*. Springer. 2010, pp. 152–165.

[21] Suzan Verberne et al. "Quantifying the challenges in parsing patent claims". In: (2010).

[22] Zhiyong Lu et al. "The gene normalization task in BioCreative III". In: *BMC bioinformatics* 12.8 (2011), pp. 1–19.

[23] Sorel Muresan et al. "Making every SAR point count: the development of Chemistry Connect for the large-scale integration of structure and bioactivity data". In: *Drug Discovery Today* 16.23-24 (2011), pp. 1019–1030.

[24] Sampo Pyysalo et al. "Overview of the ID, EPI and REL tasks of BioNLP Shared Task 2011". In: *BMC bioinformatics*. Vol. 13. 11. Springer. 2012, pp. 1–26.

[25] Claire Nédellec et al. "Overview of BioNLP shared task 2013". In: *Proceedings of the BioNLP shared task 2013 workshop*. 2013, pp. 1–7.

[26] Hidir Aras et al. "Applications and Challenges of Text Mining with Patents." In: *IPaMin@ KONVENS* (2014).

[27] Yoon Kim. *Convolutional Neural Networks for Sentence Classification*. 2014. DOI: `10.48550/ARXIV.1408.5882`. URL: `https://arxiv.org/abs/1408.5882`.

[28] WIPO. *Word Intellectual Property Organization*. 2014.

[29] Martin Krallinger et al. "Overview of the CHEMDNER patents task". In: *Proceedings of the fifth BioCreative challenge evaluation workshop*. 2015, pp. 63–75.

[30] Martin Krallinger et al. "The CHEMDNER corpus of chemicals and drugs and its annotation principles". In: *Journal of cheminformatics* 7.1 (2015), pp. 1–17.

[31] Andre Lamurias, Joao D Ferreira, and Francisco M Couto. "Improving chemical entity recognition through h-index based semantic similarity". In: *Journal of cheminformatics* 7.1 (2015), pp. 1–9.

[32] Robert Leaman, Chih-Hsuan Wei, and Zhiyong Lu. "tmChem: a high performance approach for chemical named entity recognition and normalization". In: *Journal of cheminformatics* 7.1 (2015), pp. 1–10.

[33] Yanan Lu et al. "CHEMDNER system with mixed conditional random fields and multiscale word clustering". In: *Journal of cheminformatics* 7.1 (2015), pp. 1–5.

[34] Stefan Senger et al. "Managing expectations: assessment of chemistry databases generated by automated extraction of chemical structures from patents". In: *Journal of cheminformatics* 7.1 (2015), pp. 1–12.

[35] Buzhou Tang et al. "A comparison of conditional random fields and structured support vector machines for chemical entity recognition in biomedical literature". In: *Journal of cheminformatics* 7.1 (2015), pp. 1–6.

[36] Yukun Zhu et al. "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 19–27.

[37] Saber A Akhondi et al. "Chemical entity recognition in patents by combining dictionary-based and statistical approaches". In: *Database* 2016 (2016).

[38] Raul Rodriguez-Esteban and Markus Bundschus. "Text mining patents for biomedical knowledge". In: *Drug discovery today* 21.6 (2016), pp. 997–1002.

[39] Guolin Ke et al. "Lightgbm: A highly efficient gradient boosting decision tree". In: *Advances in neural information processing systems* 30 (2017).

[40] Martin Krallinger et al. "Information retrieval and text mining technologies for chemistry". In: *Chemical reviews* 117.12 (2017), pp. 7673–7761.

[41] Stefan Senger. "Assessment of the significance of patent-derived information for the early identification of compound–target interaction hypotheses". In: *Journal of cheminformatics* 9.1 (2017), pp. 1–8.

[42] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[43] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[44] Saber A Akhondi et al. "Automatic identification of relevant chemical compounds from patents". In: *Database* 2019 (2019).

[45] Takuya Akiba et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2623–2631.

[46] Thomas Wolf et al. "Huggingface's transformers: State-of-the-art natural language processing". In: *arXiv preprint arXiv:1910.03771* (2019).

[47] Zenan Zhai et al. "Improving chemical named entity recognition in patents with contextualized word embeddings". In: *arXiv preprint arXiv:1907.02679* (2019).

[48] Iz Beltagy, Matthew E Peters, and Arman Cohan. "Longformer: The long-document transformer". In: *arXiv preprint arXiv:2004.05150* (2020).

[49] Suchin Gururangan et al. "Don't stop pretraining: adapt language models to domains and tasks". In: *arXiv preprint arXiv:2004.10964* (2020).

[50] Jiayuan He et al. "Overview of ChEMU 2020: named entity recognition and event extraction of chemical reactions from patents". In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer. 2020, pp. 237–254.

[51] Jinhyuk Lee et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4 (2020), pp. 1234–1240.

[52] Qi Liu, Matt J Kusner, and Phil Blunsom. "A survey on contextual embeddings". In: *arXiv preprint arXiv:2003.07278* (2020).

[53] Camilo Thorne and Saber Akhondi. "Word Embeddings for Chemical Patent Natural Language Processing". In: *arXiv preprint arXiv:2010.12912* (2020).

[54] Manzil Zaheer et al. "Big bird: Transformers for longer sequences". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17283–17297.

[55] Jiayuan He et al. "ChEMU 2021: reaction reference resolution and anaphora resolution in chemical patents". In: *European Conference on Information Retrieval*. Springer. 2021, pp. 608–615.

[56] Yuan Li et al. "Overview of ChEMU 2021: Reaction Reference Resolution and Anaphora Resolution in Chemical Patents". In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer. 2021, pp. 292–307.

[57] Juraj Mavračić et al. “ChemDataExtractor 2.0: Autopopulated Ontologies for Materials Science”. In: *Journal of Chemical Information and Modeling* 61.9 (2021), pp. 4280–4289.

[58] Antonio Miranda et al. “Overview of DrugProt BioCreative VII track: quality evaluation and large scale text mining of drug-gene/protein relations”. In: *Proceedings of the seventh BioCreative challenge evaluation workshop*. 2021.