Capsule Based Neural Network Architecture to perform completeness check for Patent Eligibility Process

1st Saurabh Srivastava
TCS Research
Noida, India
sriv.saurabh@tcs.com

1st Puneet Agarwal
TCS Research
Noida, India
puneet.a@tcs.com

1st Gautam Shroff
TCS Research
Noida, India
gautam.shroff@tcs.com

1st Lovekesh Vig
TCS Research
Noida, India
lovekesh.vig@tcs.com

1st Vidya Vikas
Tata Consultancy Services
Mumbai, India
vidya.vikas@tcs.com

***Abstract*—In the process of filing patents, attorneys need to ask many questions, to the inventors, to ascertain patent eligibility. We propose to ease up such conversation through a deep learning-based system. This system can automatically check whether all key ingredients required for checking the patent eligibility are present in technical write-up shared by the inventors. If not, the inventors can provide the missing information. We present a trainable model to identify various ingredients such as the objective, motivation, new observation, etc. from research articles. We model this as a sentence classification problem, which is a difficult task because a patent can be filed in any domain, and sentences involved can often be very long. To this end, we propose a dilated LSTM and capsule-based neural network architecture. We present experimental results of the proposed model on a real-world patent dataset covering patent applications in diverse domains in which our organization is carrying out research and innovation activities, and also three publicly available sentence classification datasets. Through empirical analysis, we show that a) Our model performs significantly better than several strong baselines on the patent dataset; b) Performing dilation operation on LSTMs allows us to capture long term dependencies; c) our model is comparable to existing state-of-art approaches on the publicly available datasets; d) Error analysis through LIME shows that the proposed approach can help patent attorneys to interpret the decisions taken by the classifier.**

***Index Terms*—Natural Language Processing, Capsule Networks, Dilation, Patent Eligibility, Deep Learning**

I. Introduction

The process of filing a patent application has many stages, starting with the inventors statement describing the innovation (referred to as IDF, Invention Disclosure Form). The IDF is then reviewed by patent attorneys meticulously to decide if the work is patent eligible. Attorneys at this stage assume that the inventors statements are correct, i.e., prior art search and verification is not performed. Before a prior art search is conducted to assess patentability, attorneys assess the eligibility of the innovative work. Towards making a conclusive judgement they need to identify some key *ingredients* from the IDF, such as"context", "motivation" and so on. Identifying these necessary ingredients requires a careful manual reading and verification process in which both the attorney and the inventor have to confer with each other to reach a mutual agreement on key contributions and the novelty of the invention. To identify the key contributions and the novelty of the invention, attorneys often need to read the scientific articles written by the inventors related to the invention to understand the invention and to get answers to specific questions that they need for formulating the patent documents. The attorney then walks through a decision making work-flow, e.g., USPTO guidelines for accessing patent eligibility and uses prior experience to declare whether the work is patent eligible.

It has been observed that the IDF gets rejected due to absence of key ideas. Attorneys' time is also best utilized when there is a high likelihood of the patent getting filed. A sentence classifier that can perform some basic checks, and elicit information about the key *ingredients* from the inventors before an IDF is presented to an attorney, can help to aid easy understanding of the invention from a patentability perspective and reduce legal costs. An organization can encourage inventors to file patents by interacting with the system as opposed to the traditional approach of filling an online form to capture details of their work. Sometimes even experienced inventors provide ambiguous answers, as questions regarding ingredients can be challenging. Hence, an automated system to map an inventor's statements from papers to ingredients will help reduce the time spent on these problems.

To create a classifier, we need training data that can help us determine the input type (class) for a given sentence. In the absence of a suitable publicly available training dataset, we decided to create our own. To create training data, we first designed a taxonomy of necessary ingredients required by patent attorneys to check for patent eligibility of an invention, in line with USPTO guidelines (one of the co-authors of the paper is a patent attorney who helped us in designing the taxonomy). To create training samples, we took abstract and introduction part of research articles and annotated the sentences within them.

The motivation behind using research articles as a proxy for patent filing process is based on an observation that a patent is often filed on the basis of a technical article written by the inventors or based on the users' interaction with a conversational system. In the absence of latter, we used research articles as an approximation of inventors' statement

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

and worked on the classification of sentences present in the abstract and introduction sections of research articles.

While studying the research articles we observed that novel work can belong to any domain, e.g., life sciences, material science, data science, etc., which is why the IDFs often contains new terminologies belonging to different domains introducing a bunch of Out Of Vocabulary (OOV) words. The presence of OOVs makes sentence classification harder, an example of such statements is shown in Figure 1. Further, these sentences tend to be longer than the average sentence length which requires proper treatment of long term dependencies in the input. As a result of these challenges, we observed that some of strong baseline sentence classification approaches didn't perform well on our dataset demanding a better approach to deal with OOVs and long term dependency. We will be making our dataset publicly available for future research of such statements.

To solve the aforementioned challenges, we 1) created a taxonomy (Section 2.1) to create training data and, also, 2) present a novel neural network architecture, that performs dilation [1] on Bi-LSTMs, and a capsule [2] layer to deal with OOVs and long term dependencies. We demonstrate via our proposed model that dilation can help us capture important words or phrases from large sentences and that capsules are a good alternative to attention as it helps us identify discriminative phrases. Capsule layers have mainly been used with CNNs for various image-based learning task [2] and for text applications [3]. We use an RNN module with dilation for capturing the long term dependencies and then use the capsule layer to bolster the semantic processing required for ingredient identification. Our proposed model achieves state-of-the-art performance on three publicly available datasets and outperforms benchmark approaches on our dataset.

Key contributions of this work are summarized as follows:

1) We present a novel neural network architecture that uses a capsule layer with dilated BiLSTMs to classify statements into one of the eight ingredients. To the best of our knowledge, this is the first use of dilated BiLSTMs with capsule networks.
2) We created a new dataset for identification of patent eligibility ingredients, from about $400+$ research articles, against most of these articles a patent was filed by authors. We release this dataset comprising of approx. 9000 sentences for future research. On this dataset, our model outperformed all the strong baseline approaches.
3) We analyze the challenges involved in the classification of sentences involved in the patent filing domain and scientific literature domain and demonstrate that traditional approaches do not perform well in such settings. We have also analyzed the mistakes made by our proposed model.
4) We obtained state-of-the-art results on three publicly available datasets, often used for sentence classification.

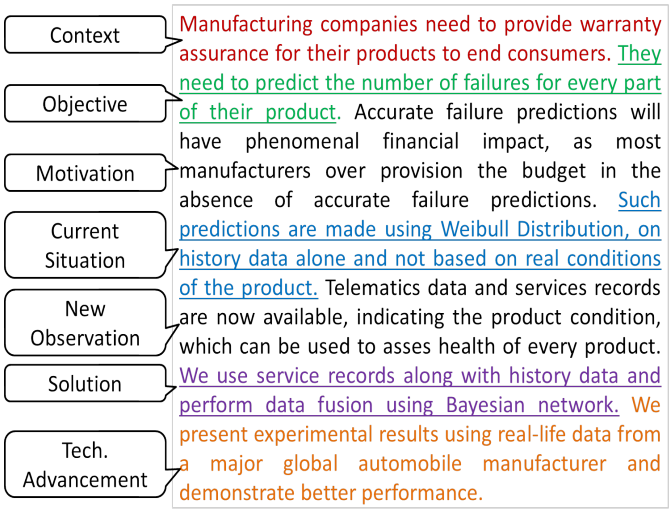

Fig. 1. Sample Invention Disclosure Statement

II. Problem and Taxonomy Description

To file a patent application, an inventor has to first submit an invention disclosure form (IDF) which contain all the necessary aspects of the inventions and the novel contributions made by the inventor. From such IDF, we need to extract the key ingredients of the invention and present them to the attorney to perform patent eligibility check. We describe these eight ingredients later in this section.

In this paper, we intend to build a classification model that can map statements of a research article to one of ingredients or classes. We can formalize this problem as a sentence classification problem in which, given a statement and a label pair corresponding to it $\langle X_i, c_i \rangle$, where X_i is a sequence of words, i.e., $X_i = \{w_i^1, w_i^2, ..., w_i^m\}$. For the sentences X_i against which the class labels are not available, we intend to predict the class c_i such that : $c_i = \text{argmax}_{c_j \in \mathcal{C}} f(c_j|X_i)$. Here, $\mathcal{C}$ is a set of all the ingredients c_j.

A. Key Ingredients of IDF

We explain the key ingredients required by attorneys to determine the eligibility of a patent. We consider a running example, shown in Figure 1, for this description.

Context or Background: Sentences that provide background information or context of the invention are marked in this category. The reader gets to know about the area of invention and a general introduction to some of the important keywords of the related domain, e.g., the first sentence of Figure 1, indicates that the invention is related to warranty of products of manufacturing companies.

Objective (OBJ): This kind of sentences describes inventors' focus of cardinal importance, i.e., what is the exact problem that the inventors solve through this invention. For example, the second sentence of Figure 1 indicates that the key focus is to predict the number of failures of a part of a product.

Motivation (MOT): Every research is supposed to somehow benefit the research or social community. The incentive to let the researcher devote their time and resources on their

work are described in Motivation (MOT) section of the article. For example, the third sentence in Figure 1 explains that the financial impact of the predictions is the key motivation.

Current Situation and its Problems (CSP): Normally, various shortcomings and flaws of the current state-of-art are studied and an improvement/alternative approach is proposed. Sentences that describe the state-of-the-art approaches, and their shortcomings, are categorized in this class. For example, the sentence written with blue color in Figure 1 indicates that currently Weibull distribution on history data is used and it does not take into account the real condition of the product.

New Observation (NO): In order to achieve the goal as given in 'OBJ', and solve the problems reported in 'CSP' inventors observe the process and the technology used very carefully, and pick an observation that becomes the basis of their novel solutions. Sentences that present the details of such observations are categorized in the 'NO' category. For example, in Figure 1 inventors observe that data from service records can be used to take into account the operating condition of the product when trying to predict the number of part failures.

Solution: Sentences that present an overview of the proposed solution are assumed to be in this category. For example, the second last sentence in our running example proposes a solution to the problem described in 'CSP'.

Technical Advancement: The competency of the novel solution is usually portrayed quantitatively, e.g., measuring accuracy or computational efficiency. We cluster all such statements into the class Technical Advancement (TA). The last sentence in Figure 1 is of similar type.

Organization (Org): Sentences of this type often occur in the research articles. For example the last paragraph of Section I in this paper. These statements are not necessary for patent eligibility test, but since our objective was to divide the introduction and abstract in a mutually exclusive way we decided to annotate them.

For defining the classes and identifying the important parts of the research articles, we conferred the patent attorneys from our organization and followed the necessary USPTO guidelines for patent eligibility test. The annotation part was carried out by 3 researchers and 1 attorney and the dataset was finalized after mutual consensus. During data creation, we identified certain traits/ cues that can be used to identify ingredients. For e.g., phrases like *is called, was introduced, known as* etc. can be used to identify Context sometimes. Similarly, *In this paper, we need to, our objective, our aim is* etc. can be used to identify Objective. Phrases like *is an important task, we need to* etc. can be used to identify Motivation, *are done using, currently is performed, has limitation, drawbacks of* to identify Current Situation and it's Problems, *is observed to, is identified, indicating that* to identify New Observation, *we propose, we used, performed experiments* to identify Solution, and *outperform, experiments show that, achieve better* to identify Technical Advancement. These words and their synonyms ***glue*** together proper context information of research article and play an important role in identifying the ingredients. We will call these words/ phrases as *glue words* in the remaining part of this paper.

III. Related Work

Basic approaches Early classification approaches involved models like Tf-Idf [4] [5], J48 [6]. These approaches involved the use of manually crafted features followed by the use of a classification model such as SVM [7]. After the emergence of the new state-of-the-art methodologies for embedding words [8] into a fixed length vector, this form of representation learning became a norm in NLP.

CNN based approaches CNNs are known for squeezing out the distinctive phrases from the text and hence are suitable for text classification [9]. However, one needs to be careful in selecting the window size of the convoluting kernels, larger the window size, larger is the parameter space, which makes them difficult to train. On the other hand, smaller window sizes may lead to loss of important information [10]. CNNs, typically involves convolving over the input data first and then performing a down-sampling method to extract only high-level features. Traditionally, Maxpool as a down-sampling method has been used as it selects only the most relevant feature from the input but loses temporal information, if present. In [1], authors showed that introducing holes in the convolutional layer can be used to obtain a better sentence representation.

RNN based approaches Another choice is to use RNNs [11], which uses the recurrent structure to capture the long term dependencies in texts and hence, can introduce less noise than CNNs. Yet, RNNs are known to be biased towards the extreme ends of the input and hence may lose information in case of large documents [10], where an important feature may appear anywhere not only at extremes. Different variants of such architectures have also been used for training, e.g., the use of the siamese network in [12] to train base layers of classification.

Advanced approaches A combination of RNN and CNN have been used to improve the results of the task by using each other's pros and cons [10], [13]. Attention has also been used along with RNN with a view to focus on specific words for classification [14], which has resulted in better classification accuracy. Recently, Capsule Networks [2] have been proposed and have improved the recognition accuracy on MNIST dataset and reduced the error rate by about 45%. Capsules have been seen performing well on some of the text classification problems as evidenced by [3], [15], [16], [17], and [18].

Patent related tasks and approaches A wide variety of research has been done in the patent domain covering the task of Claim Parsing [19] where the authors designed a parser and chunker to extract the claims from patent data, [20] extracts keywords that relate to novelties or inventive steps from patent claims using their structure. [21] proposed techniques to improve prior art search techniques which however is out of the scope of this paper.

IV. Proposed Architecture

In this section, we describe the proposed model which consists of dilation on BiLSTMs, a capsule layer and a fully connected layer.

1) Classification Model with Dilated LSTMs and Capsules : The problem is formulated as a classification problem, where model M takes as input a sentence X_i and maps it into one of the labels. Our model is composed of an *embedding layer*, a *feature extraction layer* to encode the sentence representation, a *primary capsule layer*, a *convolutional capsule layer* followed by a fully connected layer to compute the probabilities. Next, we provide details of each layer of our model.

Embedding Layer: A sentence composed of N words $[w_1, w_2, ..., w_N]$ is passed through the embedding layer which maps each word w_i to it's real-valued fixed length vector $\mathbf{v_i}$ containing it's lexical and semantic representation. The embedding layer is typically represented by a "Weight-matrix" $\mathbf{W} \in \mathbb{R}^{d_{word} \times |V|}$, where d_{word} is the vector dimension and $|V|$ is the vocabulary size. Each column j of weight matrix corresponds to a column vector $\mathbf{w_j} \in \mathbb{R}^{d_{word}}$ for the j^{th} word in vocabulary.

Feature Extraction Layer: The feature extraction layer is used to enocde the high level features into a vector $\mathbf{s_{sen}}$ obtained after processing the embedding vector of each word in sentence from the embedding layer. We have used BiLSTMs [22] for this purpose and hence, obtained $\mathbf{c_i} = [\overrightarrow{\mathbf{c_i}}; \overleftarrow{\mathbf{c_i}}] \in \mathbb{R}^{2 \times d_{sen}}$ for a word w_i where, $\overrightarrow{\mathbf{c_i}}$ and $\overleftarrow{\mathbf{c_i}}$ are right and left contexts, and d_{sen} is number of BiLSTM units. Finally, for all the N words, we have $\mathbf{C} = [\mathbf{c_1}, \mathbf{c_2}, ..., \mathbf{c_N}] \in \mathbb{R}^{N \times (2 \times d_{sen})}$ where, N is the max number of words fed to the BiLSTMs at each time step and d_{sen} is the number of LSTMs used to extract the summary. To eschew from the problem of choosing proper window-size, we replaced the CNNs with BiLSTMs to reduce the possible noise and to capture smoothened context.

Primary Capsule Layer: The Primary Capsule Layer was proposed in [3] to replace the scalar-output feature detectors of CNNs with vector-output to capture local order of words and their semantic representations. Here, we propose to use a filter $\mathbf{W^b}$ which convolves with not only the adjacent context vectors $(\mathbf{c_i}, \mathbf{c_{i+1}}, ...)$ but, also with distant context vectors $\mathbf{c_{i+dr}}$, skipping dr vectors in between, which is also referred to as dilation rate. Consequently, in spite of using RNNs, which are said to be biased for extremes [10], we are able to focus on words in between the sentence also as shown in Figure 3. Attention layer also has a similar objective, however, with routing between capsules our method outperforms attention based network, as shown in Section V.

Inspired from [3], we used a shared window with holes $\mathbf{W^b} \in \mathbb{R}^{(2 \times d_{sen}) \times d}$ where, d is the capsule dimension, convolving with the context vectors $\mathbf{c_i}$. For each context vector $\mathbf{c_i}$, we used a shared window with holes $\mathbf{W_b}$ convolving with vectors in $\{\mathbf{c_{i+dr}}\}_{i=1}^{N}$ with stride of one to get a capsule $\mathbf{p_i}$

$$\mathbf{p_i} = g(\mathbf{W^b c_i})$$

where, $\mathbf{g}$ is a non-linear squash function introduced in [2] to shrink small vectors to around 0 and larger to around 1−−larger the size of capsules, larger will be probability of presence of instantiated parameters they represent, and, $\mathbf{b_1}$ is the bias vector. After the convolution operation, we have a capsule feature map $\mathbf{P} = [\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_C}] \in \mathbb{R}^{(N \times C \times d)}$ stacked with total $N \times C$ d-dimensional capsules representing the contextual capsules.

One of the major drawbacks of using Maxpooling as a down-sampling mechanism is that it leaves behind any spatial information present in an input which could be problematic for tasks related to the text. [2] proposed the use of iterative *dynamic routing* algorithm to introduce a **coupling effect** where the agreement between lower level capsules (say layer l) and higher level capsules (say $l+1$) is maintained.

Suppose we have, "m" contextual capsules with low level features at layer l, and "n" contextual capsules at layer (l+1) then, for a capsule j at layer (l+1), we calculate it's output vector by

$$\mathbf{s_j} = \sum_{i=1}^{m} \mathbf{c_{ij}} \hat{\mathbf{u}}_{\mathbf{j|i}}; \hat{\mathbf{u}}_{\mathbf{j|i}} = \mathbf{W^s_{ij}} \mathbf{u_i}$$

where, $\mathbf{c_{ij}}$ is the coupling coefficient between capsules i of layer l to capsule j of layer (l+1) and are determined by iterative dynamic routing process, $\mathbf{W^s}$ is the shared weight matrix between the layers l and l+1. Unlike [3] we used softmax for our computations and the coupling coefficients $\mathbf{c_{ij}}$ are calculated iteratively in 'r' rounds by :

$$\mathbf{c_{ij}} = \frac{\exp(\mathbf{b_{ij}})}{\sum_k exp(\mathbf{b_{ik}})}$$

Logits $\mathbf{b_{ij}}$ which are intially same, determines how strongly the capusles j should be coupled with capsule i.

Convolutional Capsule Layer: Similar to [3], we have capsules connected to lower level capsules where we determine the child-parent relationship by multiplying the shared transformation matrices followed by, the routing algorithm. By using the shared transformation matrix we calculate the candidate parent-capsule $\hat{\mathbf{u}}_{\mathbf{j|i}}$ by,

$$\hat{\mathbf{u}}_{\mathbf{j|i}} = \mathbf{W^s_{ij}} \mathbf{u_i}$$

where, $\mathbf{u_i}$ is the child capsule and $\mathbf{W^s}$ is shared weight between capsules i and j. Finally, the coupling strength between the child-parent capsule is determined by the routing algorithm to produce the parent feature map.

Fully Connected Layer: The capsules are first flattened into a list of capsules and are then multiplied by a transformation matrix $\mathbf{W^{FC}}$ followed by routing algorithm to compute class probabilities. Proposed architecture is shown in Figure 2.

V. Experiments and System details

In this section, we first describe the datasets used for training our model, explain the details of the training process and then present empirical results obtained after comparing the proposed model with other approaches.

***Patent Eligibility Dataset*:** To the best of our knowledge, there does not exist any dataset in the public domain aligned with our proposed taxonomy of statements in the patent document.

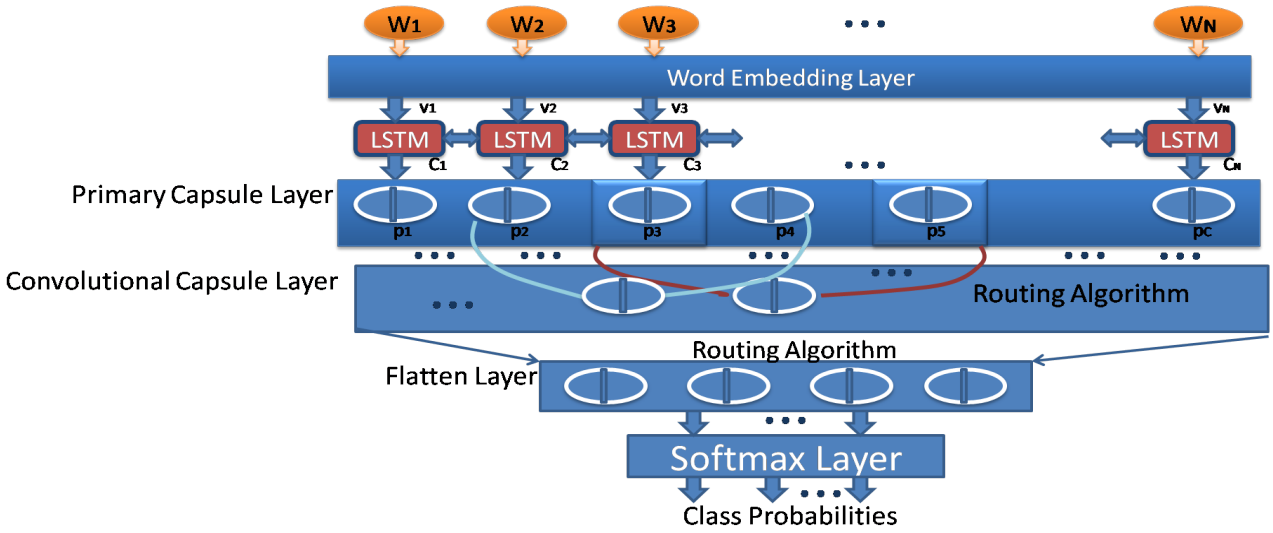

Fig. 2. Proposed Architecture

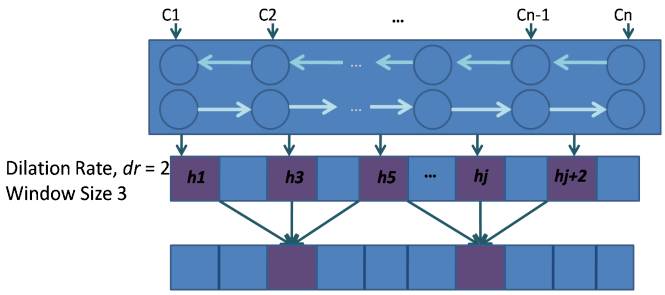

Fig. 3. Example of dilation operation

As indicated in Section II, it is observed that most of the ingredients are present in research articles. We, therefore, took 400+ research articles, against most of which patents have been filed. From the abstract and introduction of these articles, we extracted 9593 different sentences. These sentences were then annotated by 3 researchers and one patent attorney. After annotation, the dataset was divided in the ratio 90-10-10 for train, dev, and test respectively. The dataset will be released in the public domain for future research[1].

***Publicly Available Datasets*:** To compare the performance of the proposed model on the different type of sentences, we used the following datasets for comparison:

1) 20Newsgroups : The 20 newsgroups dataset is a collection of newsgroup documents. We have used bydate[2] variant of the dataset consisting of classes comp, politics, rec, and religion with standard split as specified in [10]. The 20Newsgroups dataset was chosen for testing performance of dilation operation on BiLSTMs to capture long term dependencies in large documents.
2) AG's News: AG's News is also a collection of news articles consisting of 108K train, 12k dev, and 7.6k test sentences. The average length of sentences in AG's news corpus is almost same as the length of sentences in our dataset.
3) TREC: TREC dataset was introduced for question categorization task with 6 labels, 5.4k train, .5k dev, and .5k test sentences. The average length of statements in TREC is 10 words which makes it a suitable candidate for evaluating the performance of our model on short texts.

[1]Dataset will be made public after the publication
[2]http://qwone.com/ jason/20Newsgroups/

A. Training Details

For training the proposed model, we used BiLSTMs for capturing the semantic relationship between the text components. During hyperparameter tuning, the number of LSTM units, $\mathbf{d_{sen}}$ were varied between {128-512} with a step size of 64. Number of capsules, C varied between {16-20} with their dimension d varied between {16-20}, routings, r were tested within range {3-7}. The maximum input sentence length was kept to 100 and word embeddings of dimension 300 were obtained from GloVe [23]. The dropout values were adjusted as described in [24]. All these values were tuned on our dev dataset.

B. Experiments on Patent Eligibility dataset

Baseline Methods For the performance comparison of the proposed model, we have implemented and tested some of the strong baseline models like TF-IDF, vanilla CNNs, Hybrid Channel CNNs [9], vanilla Bi-LSTMs, LSTM with skip connections, RCNN [10], CNN-LSTM [25], Feed Forward Attention Networks [26], and Capsules with CNN [3].

In our experiments, we have used the classification accuracy as a metric for the evaluation. We present a comparison of our proposed approach with the baselines approaches in Table I.

TABLE I
PERFORMANCE ON PATENT ELIGIBILITY DATASET

Models	Test Accuracy
Tf-idf	47.6
CNN	62.42
Hybrid Channel CNN [9]	66.69
BiLSTM	73.47
LSTM with Skip Connections	71.57
RCNN [10]	74.28
CNN-LSTM [25]	77.6
Feed Forward Attention Networks [26]	74.57
Capsules With CNN [3]	71.3
Our Model	**80.0**

From the results, we see that the proposed model outperform all the other models by good margins beating the second best model CNN-LSTM by approximately 3%.
From the results table, we can conclude that CNNs are outperformed by all the models using LSTMs as a feature extractor corroborating the fact that LSTMs are able to handle long sentences. Further, performing dilation helped us capture long term dependencies from input data leading us to an improvement of about 3% when compared to the second best model.

C. Error Analysis on Patent Eligibility dataset

We have selected a few misclassified samples from our test set of each class for analyzing the model weights assigned to each word. To understand the model weights assigned to input words, we used LIME [27] on each of the misclassified sentences to get a better understanding. LIME uses a local interpretable model to approximate the model in question and tries to create certain explanations of input data by performing some perturbations on input data to understand the relationship between input and output data. The output of LIME can be interpreted as weights assigned to the words where positive weights are colored in green and words with negative weights in red. LIME also provides an explanation for probabilities assigned to each class based on the weights assigned to each word. It provides an explanation for each class by assigning the positive weights (green) to the words which play a major role in assigning higher probabilities to the current class. For example, as shown from Figure 4 to Figure 10, the same sentence is explained twice to highlight the words with higher (green) and lower (red) weights for two classes (correct and predicted).
Figure 4 shows an 'Objective' misclassified as 'Current Situation', after analysis we found that the sentence contains both the information, i.e., about the problem associated with current situation (hence the positive weights for words like 'problem', 'not', etc.) and the objective of the authors (hence the positive weights for words like 'our', 'focus', etc.).
In Figure 5, we can see that a statement of type 'Current Situation' is assigned into 'Solution' class by our model. We can see clearly that the statement is describing a solution related to the author's previous proposed solution. The author has clearly described a solution but since it has been proposed earlier we decided to categorize it as 'Current Situation'.
In Figure 6, the statement is hard to dissect without proper context. If we have the knowledge that we are talking about the difference between proposed solution and some earlier proposed model then the information, that the proposed model took less time than (3-5 minutes) the other could be seen as a benefit achieved after using the proposed solution. However, without proper understanding, we can say that 'currently' it takes about 3-5 minutes for the solution to execute. So, the model needs a human understanding to classify the statement

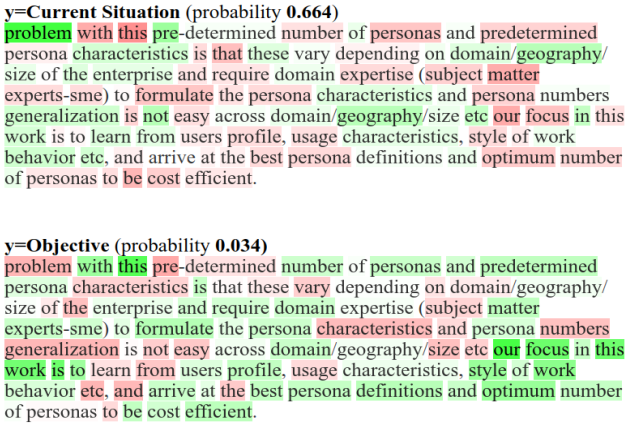

Fig. 4. Objective misclassified as Current Situation

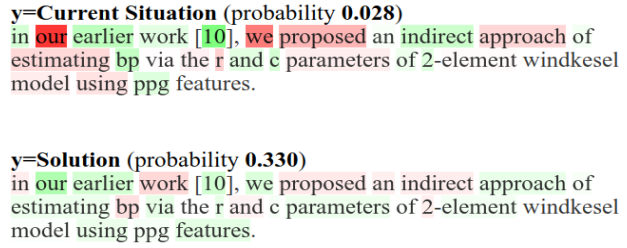

Fig. 5. Current Situation misclassified as Solution

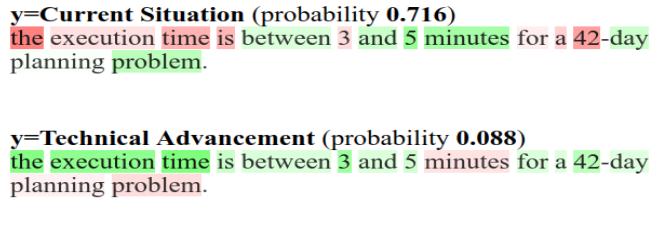

Fig. 6. Technical Advancement misclassified as Current Situation

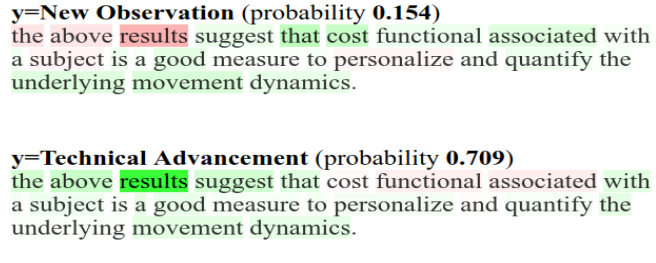

Fig. 7. New Observation misclassified as Technical Advancement

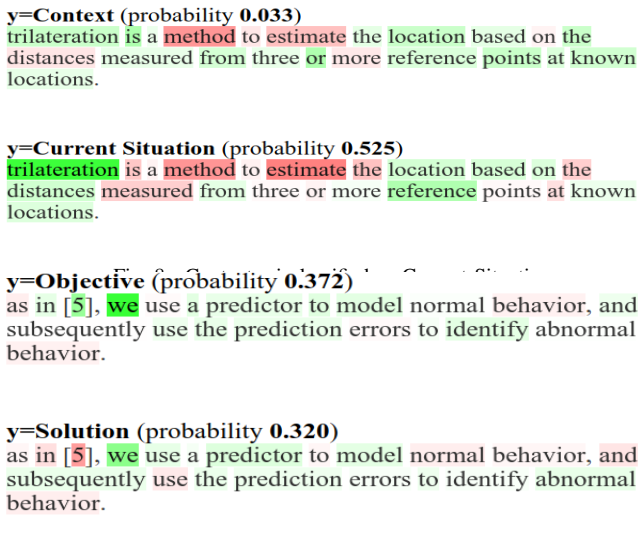

Fig. 9. Solution misclassified as Objective

here.

In Figure 7, again there is confusion as the statement is describing why something new would be good above current state-of-art. Since here the author's are describing a 'benefit' of something which is not a solution but is a good candidate for a novel solution, we decided to keep this sentence into 'New Observation'.

Figure 8 shows an example of total confusion by our model where the model was supposed to interpret the given sentence as an introduction (Context) to 'Trilateration' but instead, it misunderstood it as 'Current Situation'.

Sometimes while writing the paper, the authors use the style of "We propose a <SOLUTION> to solve <OBJECTIVE>", such statements have 'Objective' and 'Solution' amalgamated into a single sentence. Such examples are likely to confuse the model as shown in Figure 9. We can see that there is very less difference between the incorrectly predicted highest probability class (Objective) and second highest class (Solution, actual class).

Figure 10, again is an example where our model misclassifies the statement and after looking into probability scores we can see that the scores are evenly distributed in between the classes explaining the fact that the model was not sure about any class but the presence of words like 'need', 'imminent' etc. forced the model to assign higher probability to class 'Current Situation'.

Above analysis shows that the proposed model gets confused when there is a presence of multiple sources of information pertaining to different classes. Figure 8 and 10 shows that there is not enough data for classes like 'Motivation' and 'Context'. Rest of the examples show that the model or annotators should be familiar with the nuances of the statements belonging to different classes while classifying the sentences. It was always difficult while annotating these statements and hence the model confusion is likely to be reduced by a proper understanding of the document.

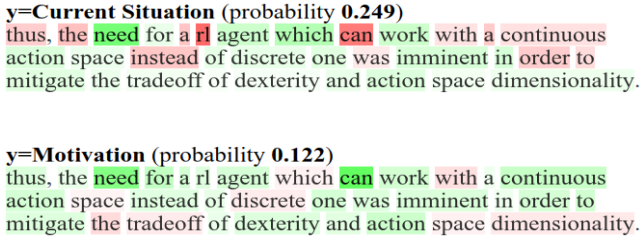

Fig. 10. Motivation misclassified as Current Situation

D. Analysis of LIME results

From the discussion provided in Section 5.3, we can see that the problem created due to the presence of OOVs and long term dependencies could be tough to handle. To interpret our model, we analyzed some of the misclassified statements from our test set and identified three more problems:

1) Presence of multiple labels: In some test samples, we found that a single statement could convey ideas from multiple classes making the problem a candidate for multilabel classification problem. However, while annotating the data, we found number of such statements to be very less (less than 500) to change the problem type from multi-class classification problem and decided to assign only one label to each statement. Further, the response given by the inventors should be clear enough to answer attorney queries as ambiguous answers will make novelty identification challenging. Restricting the problem to a multiclass classification problem enables us to restrict user input containing ambiguous answer while identifying IDF ingredients.

2) Prior understanding of inventors' approach: We can argue that having prior understanding of inventors work could help us understand the IDFs better, but as evidenced from Figure 6, these statements require a proper human understanding of the task at hand which is very tough to handle by the proposed approach.

3) Attorneys Judgement for completing the verification process: In some misclassified examples the probabilities assigned to the classes were very similar. For example, for a single statement, we can have 'Objective' and 'Solution' assigned to it with scores of .3 and .3. In such cases, it is up to the attorneys to decide to which class the statement should be mapped to. If the attorney has identified all the ingredients except 'Objective' he/ she can close the verification process *successfully* by assigning 'Objective' or, can close the verification process *unsuccessfully* by assigning 'Solution' to the current input.

E. Experiments on Public datasets

Our proposed model was also tested on several publicly available datasets like 20Newsgroup [10], AG News [28] and TREC dataset [29]. We compare the performance of our model with the best-reported accuracies on these datasets in Table II. We can see that the proposed model achieves the state-of-the-art result on 2 of the published results. It also achieves a competitive score on TREC dataset where numbers are reported in one of their CNN-non-static implementation in [3].

TABLE II
PERFORMANCE ON PUBLICLY AVAILABLE DATASET

Models	Reported Test Accuracy	Our Model
20 Newsgroups	96.69 [3]	**96.74**
AG's News	92.6 [3]	**93.4**
TREC	**93.6** [3]	92.8

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented automated identification of the ingredients for patent eligibility verification process based on the statements of the abstract and introduction of a research article given as an input to a sentence classifier. These classified ingredients provide a Patent Attorney with key information that can help him/her make a judgment on whether further analysis needs to be performed in order to determine patent-eligibility. In the future, we plan to extend our work to build a conversational agent that can ask detailed questions which would lead to a conclusion on patent-eligibility, based on prevailing guidelines provided by various statutory bodies. As mentioned in [2] there are other ways to implement the idea of capsules and as proposed in [3] we would like to investigate more deeply about the words or phrases on which capsules pay attention to in a given sentence.

VII. ACKNOWLEDGEMENTS

We would like to thanks the reviewers of this paper for thier valuable comments and all the people invlolved in this work. We would also like to thank Vipul Kapoor(vipul.kapoor@tcs.com) for his valuable contributions toward this paper.

REFERENCES

[1] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. v. d. Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," *arXiv preprint arXiv:1610.10099*, 2016.

[2] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, 2017, pp. 3856–3866.

[3] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, and Z. Zhao, "Investigating capsule networks with dynamic routing for text classification," *arXiv preprint arXiv:1804.00538*, 2018.

[4] Z. Yun-tao, G. Ling, and W. Yong-cheng, "An improved tf-idf approach for text classification," *Journal of Zhejiang University-Science A*, vol. 6, no. 1, pp. 49–55, 2005.

[5] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *Journal of Machine Learning Research*, vol. 2, no. Feb, pp. 419–444, 2002.

[6] S. Youn and D. McLeod, "A comparative study for email classification," in *Advances and innovations in systems, computing sciences and software engineering*. Springer, 2007, pp. 387–391.

[7] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[9] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[10] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification." in *AAAI*, vol. 333, 2015, pp. 2267–2273.

[11] A. Mousa and B. Schuller, "Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, vol. 1, 2017, pp. 1023–1032.

[12] P. Khurana, P. Agarwal, G. Shroff, L. Vig, and A. Srinivasan, "Hybrid bilstm-siamese network for faq assistance," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 537–545.

[13] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," *arXiv preprint arXiv:1603.03827*, 2016.

[14] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[15] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 1165–1174.

[16] S. Srivastava, P. Khurana, and V. Tewari, "Identifying aggression and toxicity in comments using capsule network," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 2018, pp. 98–105.

[17] S. Srivastava, P. Agarwal, G. Shroff, and L. Vig, "Hierarchical capsule based neural network architecture for sequence labeling," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[18] S. Srivastava and P. Khurana, "Detecting aggression and toxicity using a multi dimension capsule network," in *Proceedings of the Third Workshop on Abusive Language Online*, 2019, pp. 157–162.

[19] M. Hu, D. Cinciruk, and J. M. Walsh, "Improving automated patent claim parsing: Dataset, system, and experiments," *arXiv preprint arXiv:1605.01744*, 2016.

[20] S. Suzuki and H. Takatsuka, "Extraction of keywords of novelties from patent claims," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 1192–1200.

[21] P. Lopez and L. Romary, "Multiple retrieval models and regression models for prior art search," *arXiv preprint arXiv:0908.4413*, 2009.

[22] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[23] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," *arXiv preprint arXiv:1511.08630*, 2015.

[26] C. Raffel and D. P. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *arXiv preprint arXiv:1512.08756*, 2015.

[27] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2016, pp. 1135–1144.

[28] G. M. Del Corso, A. Gulli, and F. Romani, "Ranking a stream of news," in *Proceedings of the 14th international conference on World Wide Web*. ACM, 2005, pp. 97–106.

[29] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2002, pp. 1–7.