Content Driven Enrichment of Formal Text Using Concept Definitions and Applications

ABSTRACT

Formal text is objective, unambiguous and tends to have complex sentence construction intended to be understood by the target demographic. However, in the absence of domain knowledge it is imperative to define key concepts and their relationship in the text for correct interpretation for general readers. To address this, we propose a text enrichment framework that identifies the key concepts from input text, highlights definitions and fetches the definition from external data sources in case the concept is undefined. Beyond concept definitions, the system enriches the input text with concept applications and a pre-requisite concept graph that showcases the inter-dependency within the extracted concepts. While the problem of learning definition statements is attempted in literature, the task of learning application statements is novel. We manually annotated a dataset for training a deep learning network for identifying application statements in text. We quantitatively compared the results of both application and definition identification models with standard baselines. To validate the utility of the proposed framework for general readers, we report enrichment accuracy and show promising results.

KEYWORDS

Content Enrichment, Key Concepts, Concept Graph, Definition Extraction, Application Identification, Deep Learning

1 INTRODUCTION

Formal text is a collection of syntactic units characterized by coherency and completion, used to communicate knowledge. It is objective, unambiguous and tends to have complex sentence construction intended to be understood by the target demographic. Moreover, they contain multi-word or single-word technical terms which we call as key concepts. For example, consider an excerpt from a space and astronomy article *"What is dark matter?"*.

"Dark matter may be made of baryonic or non-baryonic matter. To hold the elements of the universe together, dark matter must make up approximately 80 percent of its matter. Most scientists think that dark matter is composed of non-baryonic matter. The candidates for this are Neutralinos, massive hypothetical particles heavier and slower than neutrinos and sterile neutrinos."

The article is intriguing but filled with key concepts such as *"baryonic matter"*, *"neutralinos"* and *"sterile neutrinos"* for which descriptions are left out. With the exception of textbooks, formal text in daily usage (for example scientific articles, blogs etc.) lacks definitions and the explanation of key concepts for the sake of brevity. In the absence of domain knowledge, it is difficult for the reader to understand these key concepts within text and their relationships which leads to an incomplete semantic understanding of the presented text. We aim to solve this problem through a content enrichment system that analyzes the input text to conditionally enrich it with information in accordance with reader's discretion. We have sourced the information from Wikipedia. Wikipedia is a collaborative and open-source medium with reliable information on general topics referenced from verifiable and notable sources. Although we equip our enrichment framework by sourcing information from Wikipedia, we propose an overall enrichment framework that can be easily extended to any available information source.

For an average reader, we want to mitigate the cumbersome problem of searching the missing information through heaps of text via high quality augmentations in the form of definitions, real-life applications [6] and inter key-concept relationships that can provide more clarity to a key concept. By relationships, we mean scenarios commonly found in text where a concept X requires prior knowledge of the concept Y which is mentioned in the text but isn't defined. The aforementioned augmentations are crucial for understanding a technical concept by any reader irrespective of his/her expertise in the corresponding domain. This belief is motivated by a set of simple questions which non-experts usually ask when they do not want in-depth knowledge but are willing to understand the concept. In other words, a non-expert might ask questions starting with "What" to understand the meaning of a concept (its definition) and "Where" to understand concept's usage or utility in the real world (its applications). Thus, we approach the problem of textual enrichment in a comprehensive way that would enable the user to make complete sense of the text.

In this work, we develop a framework for enrichment of formal text that consists of the following modules- (i) key-concepts extraction, (ii) definition identification, (iii) application identification, (iv) concept graph generation and (v) dynamic user interface (UI). The definition identification module checks for the presence of definitions for the extracted key-concepts and fetches the definitions of undefined key-concepts. The application identification module operates similarly to enrich the text with application statements. The concept graph generation module provides an overview of the pre-requisite relationships between extracted key-concepts to help the reader to understand the concept dependencies.

The main contributions of our work can be summarized as follows:

- We present a novel framework to conditionally enrich formal text with supplementary material that includes definitions, applications and concept graphs sourced from Wikipedia.
- Our method for definition and application identification utilizes LSTM networks and CNNs for sentence-level feature learning under a supervised setting.
- We created a labeled dataset for learning application statements in formal text.
- We created manually annotated datasets of formal text snippets obtained from *Scientific Articles* and lecture notes obtained from *MITOpenCourseware* to ascertain the effectiveness of our overall enrichment system.

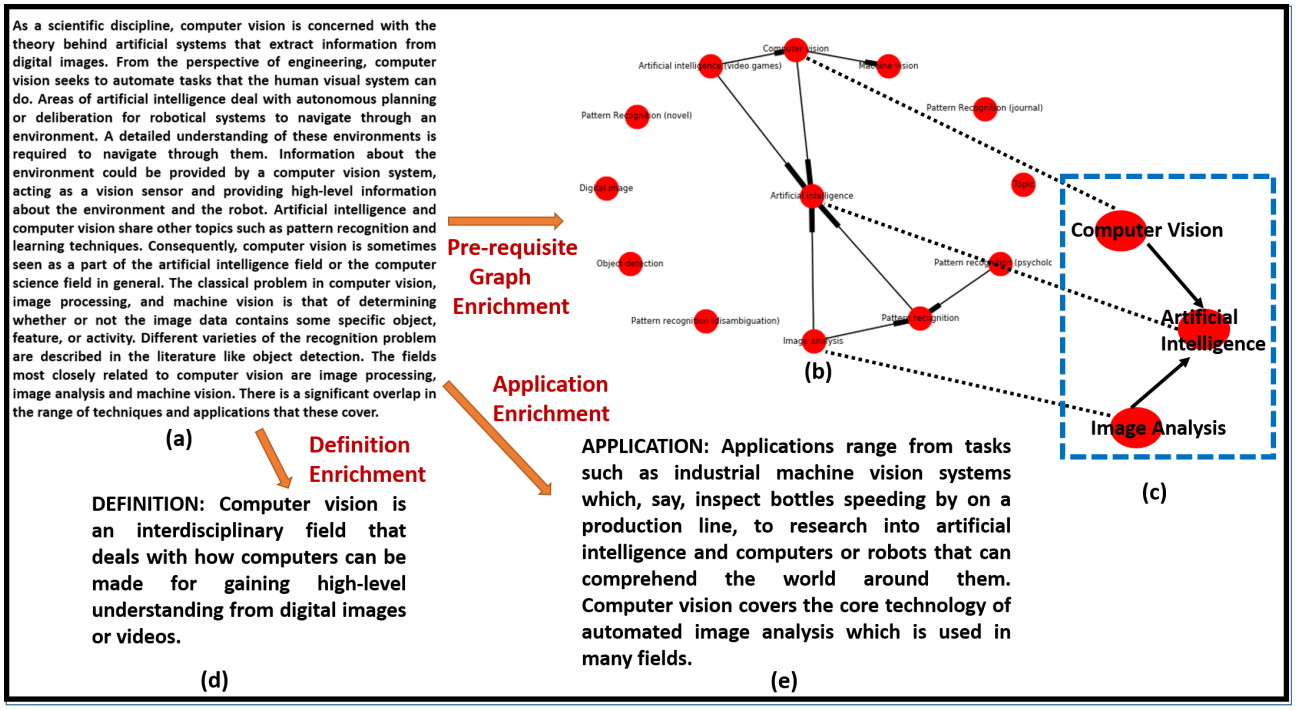

Figure 1: Output of our overall enrichment system for formal text on Computer Vision which neither contains its definition nor its application. (a) Input Text, (b) Pre-requisite graph, (c) Zoomed view of nodes and edges from Pre-requisite graph, (d) Extracted Definition, and (e) Applications of key-concept "Computer Vision" fetched by our enrichment framework.

Rest of the paper is organized as follows: Section 2 succinctly provides the related work done. Section 3 describes the overall methodology opted for enrichment. Section 4 lays down the experimental details for which results are provided and discussed and finally, Section 5 summarizes the main ideas presented in this paper and outlines the potential directions for future work.

2 RELATED WORK

One of the earliest approaches for enriching formal text, found in educational textbooks, identified key-concepts from the text, found authoritative material obtained from Wikipedia for the concepts and enriched the text with links to such authoritative material [2]. An approach was built on the work presented in [2], that identified the need for enrichment based on a decision model that used variables like syntactic complexity of writing, dispersion of key-concepts for making decisions [1]. However, the semantics of the content within the Wikipedia articles and the text being enriched is ignored by both of these works. Only the links to the articles are provided as part of the enrichment pipeline that might contain redundant content already discussed in the textbooks.

Another line of work has been explored in prior art to enrich formal text by providing a concept map generated from extracted concepts in the text. An approach is proposed in [9] that models concepts in vector space using their related concepts. An RefD score is proposed that measures the difference in the way the related concepts refer to each other and identify their pre-requisite relationship. In [5], the method utilized cross-entropy and information flow separately to infer concept dependency relations. A method that jointly optimizes the two subproblems - key concept extraction and concept relationship identification for concept map extraction is proposed in [15].

Identifying definition statements from formal text has also attracted a lot of attention in the community. Some of the earliest approaches used hand-engineered features for automatic extraction of definitions from the text. For example, in [11], an annotated dataset of definitions extracted from Wikipedia is introduced and the method proposed identifies star-patterns and word-class lattices from text for automatic definition extraction. A compendium of word level features for a weakly-supervised bootstrapping approach to classify sentences is proposed in [3]. To automate the learning of features for definition extraction, the method proposed in [8] models the problem as a supervised classification task, uses LSTM to generate these features and outperforms non Deep Learning based approaches.

The prior art partly addresses some of the challenges for enrichment of formal text. We combine some of these previous efforts and address some of their shortcomings to build a content-driven enrichment system for formal text. Specifically, we enrich the input formal text with definitions, applications and a concept graph as seen in Fig. 1. In the next section, we present the details of the proposed framework.

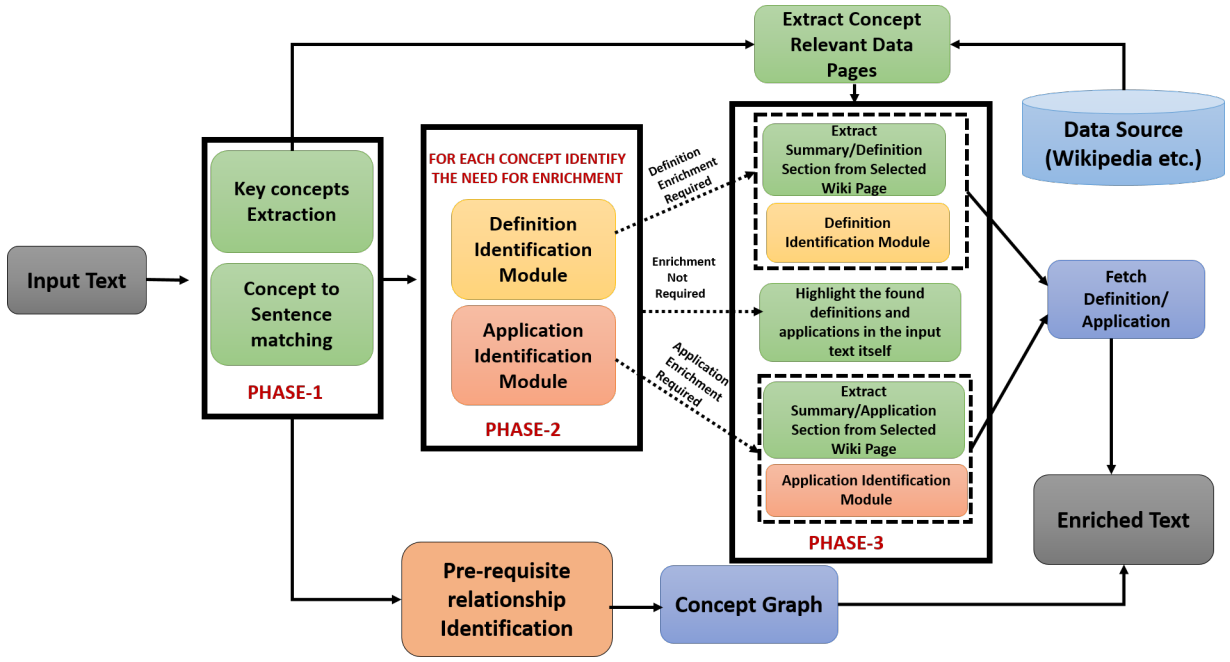

Figure 2: System Architecture of Proposed Framework

3 METHODOLOGY

Our overall methodology to enrich any input text consists of three phases as shown in Fig. 2. The initial phase extracts key concepts from the formal text, the second phase identifies the need for enrichment of these concepts, and the final phase conditionally enriches the input text with key concepts' definition and applications. This phase additionally generates a concept map from the input text which organizes key concepts based on their pre-requisite relationship with each other. We present the details of these individual system components below.

3.1 Key Concepts Extraction

Key-concepts within text exemplify the underlying idea. Extracting key concepts from text is a non-trivial task. It is not possible to always exploit trained datasets which in turn would limit the domain to work on. In this section, we present a generic pipeline to reliably extract them from input text.

3.1.1 Tagging and Linguistic filtering. The input text is first POS tagged using Stanford POS tagger [14] which assigns unique part of speech tag to every word. Tagging is needed by linguistic filters that permit only specific strings for extraction without which strings such as *of the*, *is a* will also be extracted. From [7], we identify that technical terms(key concepts) are made of nouns, adjectives and sometimes prepositions; thus we use following filters [2] - (i) P1 = C^*N, (ii) P2 = $(C^*NP)^?(C^*N)$ and (iii) P3 = A^*N^+ where N refers to a noun, P a preposition, A an adjective, and $C = A|N$.

Examples of the concepts extracted from P1 include "probability distribution function", "Fibonacci numbers", etc. Examples of P2 include "runtime of binary search", "shortest paths in graphs" while those of P3 include "long long integers", "directed acyclic graph". The use of regular expressions ensures maximal pattern matching and thus we get "directed acyclic graph" as a candidate instead of "acyclic graph" or "graph".

3.1.2 Pruning using BBC Data Corpus. Obtained set of candidate concepts from previous step might contain noun phrases which are very common that do not qualify as "technical" or as key concepts. Therefore, we leverage the word count dictionary of 90 million words BBC corpus [13]. Incorporating a corpus of common words serves to avoid the extraction of strings that are unlikely to be concepts and also handles poorly formed text; thus improving precision. This corpus was developed to identify jargon in any text based on word frequency. We chose this corpus specifically because it is an up-to-date representation of general-science related vocabulary. With its help, we identify stop-list of words which are not expected to occur as key concepts and prune the obtained concepts. Some examples include: good, day, voting, state, please, etc.

3.1.3 Stack Exchange tags based Pruning. Candidate concept extraction done so far is instrumental in the elimination of common words and phrases that are unlikely to be key concepts. However, it is essential to further filter candidate concepts such that they pertain to technical key concepts that may occur in formal text and require enrichment. In order to do so, we propose using a corpus of such technical terms. We construct this corpus using "tags" from StackExchange. Tags in StackExchange are used to annotate questions with a specific key concept that those questions pertain to. Thus, essentially a StackExchange tag refers to a Candidate concept. The StackExchange API allows us to run queries to extract tagNames from various domains that we would expect our formal text to correspond to such as Computer Programming, Chemistry, Physics, Computer Science, Mathematics etc. We can then further

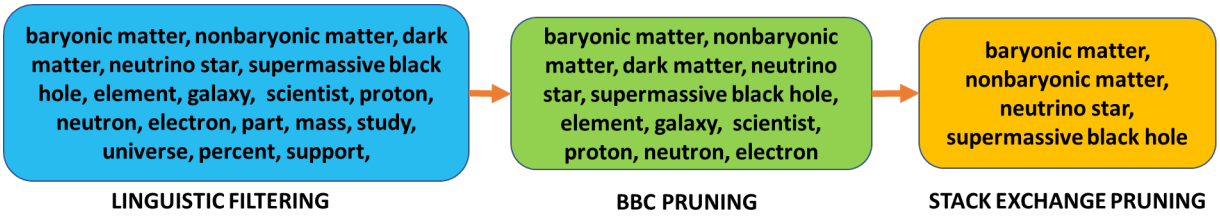

Figure 3: Key concept extraction output after every stage for the sample text shown in Table 1(a).

prune our candidate concepts by checking if they are present in our stack exchange corpus. Such pruning allows us to extract Technical concepts efficiently. The Corpus approximately contains 60,000 tags from fields such as Stack overflow, Mathematics, Physics, Information Security, Electrical Engineering, Software Engineering Chemistry, Biology, Signal Processing etc. We illustrate the above stages in Fig. 3.

Ultimately, we obtain a set of pruned concepts, $C = \{c_1, ..., c_N\}$ of N key-concepts, for which the need of enrichment is determined in the next phase.

3.2 Key Concept to Sentence Matching

Sentences quoting key-concept's definitions, applications and examples have key-concepts as their subjects. Hence, after key concepts extraction, we pre-process the text by ensuring unique association i.e. one-to-one mapping between sentences and key-concepts for subsequent identifications. For example, *"The laws of thermodynamics define fundamental physical quantities (temperature, energy, and entropy) that characterize thermodynamic systems at thermal equilibrium."* has "*laws of thermodynamics*" as its subject since the sentences is its definition but not of key-concept "*thermal equilibrium*" which is merely mentioned in the sentence. For this, a set of sentences, S_i for every key-concept, c_i is created which contains all the sentences from the input text that have c_i as their subject using Stanford's Dependency Parser [4].

3.3 Application Identification

We formulate the Application Identification phase as a supervised binary classification problem. For every key-concept $c_i \in C$, we introduce a binary valued variable $a_i \in \{0, 1\}$, where $a_i = 1$ denotes that there exists an application for the key-concept, c_i in the input text. To learn this, we determine whether any sentence in S_i possesses a certain structure that marks the existence of concept's application. Instead of hand-engineering these patterns, we employ Neural Networks to learn them from a carefully annotated dataset. There exists myriad networks which can produce sentence embeddings in a supervised setting. A natural choice to learn sentence feature would be LSTM but we specifically use CNN along with LSTM because it excels at learning the spatial structure in input data. Our application dataset has a one-dimensional spatial structure in the sequence of words and the CNN should be able to pick out invariant features from the positive samples. These learned spatial features may then be learned as sequences by an LSTM layer.

Hence, we have the following methodology that is executed on every candidate sentence in S_i to determine a_i for given c_i:

(1) **Word Embeddings**: We encode Top-N frequent words occurring in a sentence as real-valued vectors in a high dimensional space. For this purpose, we utilize publicly available 300-dimensional GLOVE embeddings [12].
(2) **Sentence Embeddings**: Having obtained word-level embeddings in the embedding layer, we add a one-dimensional CNN and max pooling layer after the Embedding layer which then feeds the consolidated features to the LSTM.
(3) **Classification**: Ultimately, LSTM feeds the learned sentence embedding to a dense network with logistic regression classifier which then predicts labels of sentences in S_i. If any sentence is classified as an application of some key-concept, we set $a_i = 1$. This overall learning is done on a carefully handcrafted dataset for which details are provided in Section 4.1.

3.4 Definition Identification

For definition identification, we keep an indicator variable $d_i \in \{0,1\}$ for every c_i in the input text, where $d_i = 1$ shows the presence of concept's definition in set S_i. Similar to a concept's application, here too we assume that definition of a concept also conforms to pre-defined lexico-syntactic patterns. Hence, we can use Neural Networks to learn these patterns.

Similar to previous section's methodology, we employ a CNN-LSTM network to learn sentences' feature vectors for classification. Note that [8] worked on similar lines by using LSTM but their end goal was to identify definitional sentences but our end goal is enrichment, to which Definition and Application identification modules belong.

3.5 Enrichment

After identification of key concepts which require additional material, we mine Wikipedia's content in specific ways to enrich the input text. We first provide the user with a pre-requisite relationship based concept map for better understanding of hierarchy present amongst identified key-concepts. This is one of the three ways with which input text is enriched. The two other ways for enrichment are based on the results of previous section i.e. on a_i and d_i for every key-concept c_i.

Pre-requisite Relationship Identification: We define the "pre-requisite structure" for a corpus as a graph, where nodes are key-concepts to comprehend, and a directed edge A $\rightarrow$ B corresponds

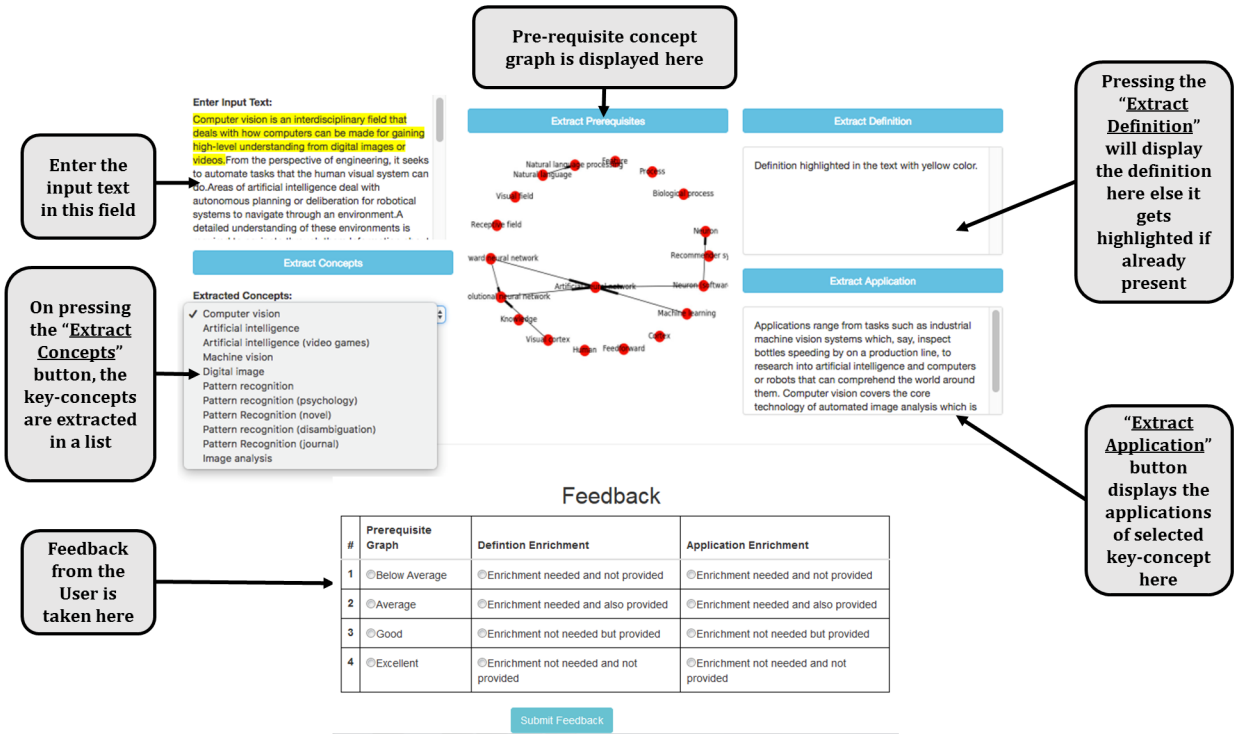

Figure 4: User Interface for the Textual Enrichment System

to the assertion that âĂIJunderstanding A is a prerequisite to understanding BâĂİ. We identify the pre-request relationship between two key-concepts A and B by equally weighing the sum of the following similarity measures:

(1) **RefD score** [9]: It is based on the inlink and outlink structure of a key-concept's Wikipage, RefD measures the prerequisite relationship between A and B. This metric is based on the idea that if most related concepts of A refer to B but few of B to A, then B is more likely to be pre-requisiste of A.

$$RefD(A,B) = \frac{\sum_{c_i \in C} r(c_i,B).w(c_i,A)}{\sum_{c_i \in C} w(c_i,A)} - \frac{\sum_{c_i \in C} r(c_i,A).w(c_i,B)}{\sum_{c_i \in C} w(c_i,B)} \quad (1)$$

where, C is the concept space, $w(c_i, A)$ is the weights importance of c_i to A and $r(c_i, A)$ is an indicator showing whether c_i refers to A. For $w(c_i, A)$, we have used equal weights i.e. $w(c_i, A) = 1$, if c_i lies in set of concepts linked from A, else $w(c_i, A) = 0$. The RefD score lies in $[-1, 1]$ where a threshold, $\theta = 0.02$ is chosen to determine the direction and existence of edge between A and B.

(2) **Wikipedia link based Semantic Similarity** [16]: This metric measures the semantic relatedness between A and B using the idea that if two concepts occur on the same page, they are more likely to be related to each other.

$$WSS(A,B) = 1 - \frac{log(max(I_A, I_B)) - log(I_A \cap I_B)}{log|C| - log(min(I_A, I_B))} \quad (2)$$

where, I_A is the set of Wikipedia concepts which link to A (inlinks) and $|C|$ is the set of all Wikipedia Concepts. This gives a score between $[0, 1]$ where higher the score means higher the relatedness between the two concepts.

Enrichment with Definitions and Applications: For definitional enrichment, we deploy our definition identification module on the set of sentences which is created from the first paragraph (summarized section) of key-concept's Wikipedia page and identify those sentences which qualify as concept's definition. Similarly, we run our application identification module.

4 RESULTS AND DISCUSSION

4.1 Tasks and Datasets

Definition Identification Model Training Dataset: We used the dataset provided by [10] to train our CNN-LSTM network for Definition Identification task. The dataset consists of 1,908 definitional sentences and 2,711 non definitional sentences created from Wikipedia which consists of domain-independent samples to prevent any kind of bias during learning. This makes the dataset apt for our purpose of enrichment of formal text because of its eligibility for domain-independent use.

Application Identification Model Training Dataset: We manually created the annotated dataset from Wikipedia which consists of 3,000 positive candidates and 3,702 negative candidates for Application Identification task. We identified generic patterns within sentences which were classified as applications of key-concepts. Every positive candidate consists of (i) the key-concept being applied,

Table 1: Excerpts from sample formal texts with annotation for some of the extracted key concepts (shown in bold).

Text
(a) Studies of other galaxies in the 1950s first indicated that the universe contained more matter than seen by the naked eye. Support for dark matter has grown, and although no solid direct evidence of dark matter has been detected, there have been strong possibilities in recent years. The familiar material of the universe, known as **baryonic matter**, is composed of protons, neutrons and electrons. Dark matter may be made of baryonic or **non-baryonic matter**. To hold the elements of the universe together, dark matter must make up approximately 80 percent of its matter. The missing matter could simply be more challenging to detect, made up of regular, baryonic matter. Potential candidates include dim brown dwarfs, white dwarfs and **neutrino stars**. **Supermassive black holes** could also be part of the difference. But these hard-to-spot objects would have to play a more dominant role than scientists have observed to make up the missing mass, while other elements suggest that dark matter is more exotic.
(b) **Risk tolerance** is a crucial factor in personal financial decision making. Risk tolerance is defined as individuals' willingness to engage in a financial activity whose outcome is uncertain. Behavioral economics is primarily concerned with the bounds of rationality of economic agents. Behavioral models typically integrate insights from psychology, neuroscience and **microeconomic theory**. The study of behavioral economics includes how market decisions are made and the mechanisms that drive public choice. **Behavioral economics** has also been applied to inter temporal choice. **Inter temporal choice** is defined as making a decision and having the effects of such decision happening in a different time.

Key Concepts	Definition	Application
Baryonic matter	Yes	No
Supermassive black holes	No	No
Behavioral Economics	No	Yes
Risk Tolerance	Yes	Yes
Inter temporal Choice	Yes	No
Microeconomic Theory	No	No

(ii) a verb phrase showing how the key-concept is being applied and (iii) the field where the key-concept is being applied. For example, consider the following sentences:

(1) [*Scenery generators*]$_{concept}$ [are commonly used in] [*movies, animations and video games*]$_{fields}$.
(2) [*The COS cell lines*]$_{concept}$ [are often used by] [*biologists when studying the monkey virus SV40*]$_{field}$.
(3) [*In the production of semiconductor materials and devices,*]$_{field}$ [*octafluorocyclobutane*]$_{concept}$ [serves] as a deposition gas and etchant.

The sentences show how different key-concepts are applied in their corresponding fields. We identified such sentences from Wikipedia and manually annotated them to create the dataset. This was done by one of the authors and reviewed by the rest. With careful linguistic analysis, some exceptions to general trend were also found:

(1) Sentences which state there are no existing applications of concept: These type of sentences dismiss the need for enrichment which makes them positive candidates for application identification otherwise the module will look for an application which doesn't exist; e.g. *Molecule-based magnets currently remain laboratory curiosities with no real world applications, largely due to the very low critical temperature at which these materials become magnetic.*
(2) Uninformative Sentences: These category of sentences qualify as an application syntactically but provides no information. e.g. *Vinyl banners have many uses and applications.*
(3) Definitional applications: Usually, while creating the dataset we refrained from incorporating sentences which are definitions of some key concepts as positive samples. But sometimes a key-concept's definition is its application like "*A dashpot is a common component in a door closer to prevent it from slamming shut*"; the sentence is both "*dashpot's*" definition and application.

System Evaluation Dataset[1]: To evaluate the overall effectiveness of our enrichment system, we created following Ground Truth datasets: (1) Lectured notes from MIT Open Courseware on 'Physics','Chemistry','Algebra' and 'Algorithms'. They contain a total of 80 educational texts containing an average of 15 pages each which further consists of 10-15 key concepts each page. We also created (2) 'Articles' dataset which consists of 100 articles from multitude of science magazines that has 15-20 key-concepts each. Excerpt from some sample text is shown in Table 1. Lecture notes datasets have significant number of defined concepts in topics such as 'Probability', 'Chemical Reactions' etc but lacks their real-life applications and thus is suitable to test our application identification module effectively. On the contrary "Articles" dataset is rich with formal texts targeted for specific demographics; thus, lacks definitions(explanations) of significant number of concepts.

4.2 Evaluation Metrics

Model Learning: We evaluated the performance of all the learned models during 10-fold cross validation on definitions' and applications' dataset using Precision, Recall, F1-measure and Validation Accuracy.

[1]We plan on releasing the Application and System Evaluation dataset in the future.

Enrichment System: To evaluate the performance of our enrichment system, we have the following metrics:

- **Key Concept Extraction (KCE)** : KCE, the first phase of our enrichment system is evaluated using using usual notions of Precision and Recall.
- **Overall Enrichment Accuracy (EA)**: Identification and Extraction phase is collectively evaluated using EA which is calculated using following notions of True/ False Positives/Negatives.

$$EA = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where, TP is the total number of cases where enrichment was needed for a particular key concept and the system provided it and TN is the total number of cases where the key-concept didn't require enrichment i.e. any additional information and the system identified the important information already present in the text. We have similar notions for FP and FN which are cogently shown in Table 2.

		Ground Truth	
		Enrichment Required	Enrichment Not Required
Proposed Method	Enrichment Provided	TP	FN
	Enrichment Not Provided	FP	TN

Table 2: Notions for Enrichment Accuracy metrics

4.3 Experiment Settings

Training : We trained base line models and the CNN-LSTM model used in our enrichment framework separately on both datasets using 10-fold cross validation to come up with the best hyper-parameter settings. We restricted the Definition dataset to Top-1000 and our Applications dataset to Top-5000 frequent words. These top frequent words of every sentence were encoded into 300-Dimensional GLOVE vector embeddings which constituted the embedding layer. Following are the architectural details of different models stacked on top of the embedding layer:

- **LSTM**: Vanilla LSTM layer (h=300 memory units) → Dropout Layer,DL(dropout probability, p=0.2).
- **CNN 2-Layered**: Convolution Layer, CL(mask size=5, filter maps=128) → Max-Pooling Layer, ML(mask size=2) → CL(5, 64) → ML(2)
- **CNN 3-Layered**: CL(5, 128) → ML(2) → CL(5, 128) → ML(2) → CL(5, 64) → ML(2)
- **CNN-LSTM**: CL(5, 128) → ML(2) → LSTM layer (h=300) → DL(p=0.2).

The CNN layers used rectifier linear units for activation and since we are doing binary classification, we used Dense Layer with single neuron and sigmoid activation for every model. To avoid over-fitting of the LSTM layer, a dropout layer was added. For training, we used Adam Optimizer to minimize log loss. A batch size of 32 was chosen, vanilla-LSTM model was trained for a total of 3 epochs, CNN based models for 10 epochs and CNN-LSTM for a total of 5 epochs. Having identified the best hyper-parameter settings, we used the complete dataset to retrain the model to be used for our overall enrichment pipeline.

Overall Testing : For every sample text key concepts are extracted, followed by concept to sentence matching. All the identification models are run on S_i for every key concept, c_i to set d_i,a_i and obtain concept-dependency graph. After identifying type of enrichment (Definition,Application) required for every concept, corresponding model is run on concept's Wikipage to mine relevant information, if present. In case, the information already exists in the sample text, it is highlighted.

User Interface: We developed a User Interface(UI) shown in Fig. 4 to showcase the results of our enrichment system. The UI takes textual input from the user and identifies the key-concepts present in the text which the user can select. The enrichment system highlights the definitions and applications if they are present in text for the selected key-concept, otherwise displays them separately. Optionally, the user can also provide feedback for the results.

Any method which enriches an input text with more supplementary material should not hamper the overall comprehension and reading experience of a user. Therefore, one way opted by us is by providing definitions and applications of key-concepts separately which user can refer for better understanding of the input text. Other way could be to enrich the input text by carefully placing the application and definition of a key-concept in the input text itself; but this would require semantic understanding of the text for accurate placement of definitions and applications as well as ensuring coherency and completeness so that augmented text is not out of context. This we shall address in the future.

4.4 Results

We report the results of all the key-concepts extraction phases when implemented sequentially in Table4. Linguistic filtering is not 100% effective in recalling all the key-concepts from ground truth because of maximal string matching. For example, 'isotherm' as a concept is ignored as it is part of the string 'validity of freundlich isotherm' which gets passed through linguistic filters. BBC pruning then improves precision, but we find there is a small decrement in recall because of pruning of concepts like "Moore Voting" as it has a common(frequent) word 'Voting' in it. We observe that the inclusion of StackExchange tags increases the precision due to the richness in terms of technical concepts they provide, however, we extract some additional unneeded concepts that may be technically sound but not relevant to the context of the text thus leading to increasing cognitive burden, redundancy and irrelevant data. This trade off however depends on how well the StackExchange tag corpus is related to the context of the text, and in most common cases would lead to better results. In wikipedia based recall, we retain only those concepts which have corresponding Wikipedia articles. It is not part of our Key-concept extraction phase. However, using it significantly increases the precision but simultaneously we also observe a decrease in recall due to unavailability of concepts' Wikipages. To showcase the effectiveness of stackexchange pruning,

Model	Prec(%)		Recall(%)		F1(%)		Accuracy(%)	
	App^n	Def^n	App^n	Def^n	App^n	Def^n	App^n	Def^n
LSTM	**88.09**	92.78	82.40	88.73	84.72	90.39	86.81	92.51
CNN 2-Layered	84.42	92.05	83.17	**92.79**	83.70	92.40	85.51	93.93
CNN 3-Layered	85.55	92.25	81.13	91.24	83.70	91.65	85.25	93.40
CNN-LSTM	87.21	**93.56**	**83.73**	92.25	**85.31**	**92.83**	**87.15**	**94.33**

Table 3: Performance of different models trained on Application Identification and Definition Identification Datasets

Pruning Phase	Prec(%)		Recall(%)		F1(%)	
	Edu	*Art*	*Edu*	*Art*	*Edu*	*Art*
LF	19.44	13.23	90.34	92.59	31.99	23.15
BBC Pruning	26.20	19.1	87.75	90.1	40.36	31.53
StackExchange	53.20	32.38	81.37	82.92	64.34	46.57
Wikipedia Recall	78.68	72.04	76.35	81.71	77.50	76.57

Table 4: Performance of subsequent phases of concept extraction

Datasets	Definition(%)	Application(%)
Education	77.17	75.14
Articles	78.57	81.81

Table 5: Enrichment Accuracy : Performance of our complete end-to-end model on both Educational Text and Article Datasets

we computed the results for Wikipedia based pruning without it: $Prec$: 75.81%, $Recall$: 75.46% and $F1$: 75.64%.

In Table 3 we present the results of different Application and Definition identification models. It is quite evident from the results that the CNN-LSTM model preformed better as expected on both the datasets with less number of parameters and faster training time than vanilla LSTM model.

Finally, performance of individual components of our enrichment system on both system evaluations datasets are reported in Table 5. In our proposed methodology, Wikipedia information extraction phase is dependent on enrichment requirement identification phase. Thus, false positives, FP are due to- (i) concepts for which identification models predicted wrong, (ii) concepts for which identification model predicted right but extraction phase couldn't provide supplementary information.

For qualitative analysis, Fig. 1 represents the output of our enrichment system for formal text on "Computer Vision". The text neither contains its definition nor its applications, our system realizes this need and fetches them. Also, our system provides a concept graph to visualize the dependencies existing amongst key-concepts like understanding of "Computer Vision" is crucial before understanding "Artificial Intelligence". Altogether, this confirms that we have an adept system for enrichment of formal texts.

5 CONCLUSION

We proposed a novel framework to enrich formal text with supplementary material. In the proposed approach, we extract key-concepts from text, identify the enrichment need using Deep Learning and finally enrich the text with definitions, applications and a pre-requisite concept graph to make the comprehension of the text easier. We also prepared the System Evaluation Dataset and Applications dataset with in-depth linguistic analysis.

Lastly, we have done a quantitative analysis of the enrichment results to measure the effectiveness of our proposed framework. In future, we would like to validate the effectiveness of our enrichment framework on users with varying expertise by conducting a user study. We also plan to make our enrichment system produce integrated outputs where definitions and applications are stitched together with the input text ensuring coherency and completeness.

REFERENCES

[1] Rakesh Agrawal, Sreenivas Gollapudi, Anitha Kannan, and Krishnaram Kenthapadi. 2011. Identifying enrichment candidates in textbooks. In *WWW*. ACM.

[2] Rakesh Agrawal, Sreenivas Gollapudi, Krishnaram Kenthapadi, Nitish Srivastava, and Raja Velu. 2010. Enriching textbooks through data mining. In *DEV*. ACM.

[3] Luis Espinosa Anke, Horacio Saggion, and Francesco Ronzano. 2015. Weakly supervised definition extraction. In *RANLP*.

[4] Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*.

[5] Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully Burns. 2016. Modeling Concept Dependencies in a Scientific Corpus.. In *ACL*. ACM.

[6] Clyde Freeman Herreid. 1994. Case Studies in Science–A Novel Method of Science Education. *Journal of College Science Teaching* 23, 4 (1994).

[7] John S Justeson and Slava M Katz. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural language engineering* 1, 1 (1995), 9–27.

[8] SiLiang Li, Bin Xu, and Tong Lee Chung. 2016. Definition Extraction with LSTM Recurrent Neural Networks. In *CCL*. Springer.

[9] Chen Liang, Zhaohui Wu, Wenyi Huang, and C Lee Giles. 2015. Measuring Prerequisite Relations Among Concepts.. In *EMNLP*. ACL.

[10] Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *ACL*. ACM.

[11] Roberto Navigli, Paola Velardi, Juana María Ruiz-Martínez, et al. 2010. An Annotated Dataset for Extracting Definitions and Hypernyms from the Web.. In *LREC*.

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. ACL.

[13] Tzipora Rakedzon, Elad Segev, Noam Chapnik, Roy Yosef, and Ayelet Baram-Tsabari. 2017. Automatic jargon identifier for scientists engaging with the public and science communication educators. *PloS one* 12, 8 (2017).

[14] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL*. ACL.

[15] Shuting Wang, Alexander Ororbia, Zhaohui Wu, Kyle Williams, Chen Liang, Bart Pursel, and C Lee Giles. 2016. Using prerequisites to extract concept maps fromtextbooks. In *CIKM*. ACM.

[16] Ian H Witten and David N Milne. 2008. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. (2008).