INTRODUCTION TO NLP

Team : 62

Members:

Amitabh Paliwal (2022201030)
Prateek Pandey (2022201035)

Project

Extracting Keyphrases and Relations from Scientific Publications

Project Desciption

Extracting keyphrases and relations from scientific publications involves automatically identifying the most important words or phrases and the relationships between them that summarize the main topics or concepts discussed in a given scientific publication. This task is important because it can help researchers and readers quickly understand the main content and contributions of a publication, as well as enable more efficient information retrieval and literature search.

Materials Referred

- Keyphrase Extraction from Scholarly Articles as Sequence Labeling using Contextualized Embeddings (https://arxiv.org/pdf/1910.08840.pdf)
- Keyphrase Extraction with BERT Transformers and Noun Phrases(https://towardsdatascience.com/enhancing-keybert-keyword-extraction-results-with-keyphrasevectorizers-3796fa93f4db)
- Learning Rich Representation of Keyphrases from Text(https://arxiv.org/pdf/2112.08547.pdf)
- Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer(https://jmlr.org/papers/volume21/20-074/20-074.pdf)
- Automatic Keyphrase Extraction: A Survey of the State of the Art(https://aclanthology.org/P14-1119.pdf)

Datasets Used

- Semval 2010-SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles: This dataset was created for the SemEval-2010 Task 5 challenge and contains annotated keyphrases for 244 scientific articles from various domains.

- Semval 2017-ScienceIE 2017 dataset: It consists of 500 journal articles evenly distributed among the domains Computer Science, Material Sciences and Physics. Three types of documents are provided: plain text documents with sampled paragraphs, brat .ann standoff documents with annotations for those paragraphs and .xml documents with the original full article text. The training data part of the corpus consists of 350 documents, 50 are kept for development and 100 for testing
- Inspec- Inspec consists of 2,000 abstracts of scientific journal papers from Computer Science collected between the years 1998 and 2002. Each document has two sets of keywords assigned: the controlled keywords, which are manually controlled assigned keywords that appear in the Inspec thesaurus but may not appear in the document, and the uncontrolled keywords which are freely assigned by the editors, i.e., are not restricted to the thesaurus or to the document. In our repository, we consider a union of both sets as the ground-truth..

Models Used

Successfull Models

Statistical Models

- TF-IDF (Term Frequency-Inverse Document Frequency): This approach is a statistical method that ranks the importance of each word or phrase based on its frequency in the paper and its rarity in the overall corpus of scientific papers. The TF-IDF score for a word or phrase is calculated as the product of its term frequency (TF) in the paper and its inverse document frequency (IDF) in the corpus of scientific papers. Words or phrases with high TF-IDF scores are considered more important and relevant to the paper.
- The team used TF-IDF transformers, which involved converting text into multiple N-grams ranging from 1-n. We chose n = 3
- Then Using Count Vectorizer converted the text data into numbers
- Then, for all the features in the given text, TF-IDF frequency was calculated using TF-IDF transformers.
- The top N (N = 15) features were then extracted, having the maximum TF-IDF frequency.

Neural Models

Bilstm-CRF

- The approach involves representing the words in the input text using deep contextualized embeddings and using a Bidirectional LSTM-CRF for sequence labeling. The proposed

architecture is evaluated on three benchmark datasets, namely Inspec, SemEval 2010, and SemEval 2017, using both normal and pretrained glove word embeddings.

- The Bidirectional LSTM-CRF is a popular choice for sequence labeling tasks, as it is able to capture both forward and backward dependencies in the input text. The CRF layer also allows the model to enforce global constraints on the output sequence, ensuring that the predicted keyphrases are coherent and consistent.
- The evaluation of the proposed architecture on three benchmark datasets allows for an objective comparison of its performance against existing approaches. The use of both pretrained glove embeddings and learned embeddings provides insight into the impact of context on the keyphrase extraction task.

- Bilstm-CRF without glove(Semval 2017)-
    - Model Architecture-

```
BiLSTMCRF(
  (sent_vocab): Vocab()
  (tag_vocab): Vocab()
  (embedding): Embedding(3937, 300)
  (dropout): Dropout(p=0.5, inplace=False)
  (encoder): LSTM(300, 300, bidirectional=True)
  (hidden2emit_score): Linear(in_features=600, out_features=5, bias=True)
)
```

    - Model Results

```
{'I-Process': 6, 'I-Material': 5, 'I-Task': 7, 'B-Process': 4, 'B-Task': 3, 'B-Material': 2, '<PAD>': 1, 'O': 8, '<UNK>': 0}
```

	precision	recall	f1-score	support
2	0.18	0.23	0.20	625
3	0.01	0.53	0.02	15
4	0.06	0.10	0.07	378
5	0.28	0.25	0.26	1394
6	0.94	0.85	0.89	19591
7	0.00	0.00	0.00	0
8	0.01	0.21	0.01	28
accuracy			0.78	22031
macro avg	0.21	0.31	0.21	22031
weighted avg	0.86	0.78	0.82	22031

- Bilstm with CRF with glove(Semval 2017)
    - Architecture

```
BiLSTMCRF(
  (sent_vocab): Vocab()
  (tag_vocab): Vocab()
  (embedding): Embedding(3937, 300)
  (dropout): Dropout(p=0.5, inplace=False)
  (encoder): LSTM(300, 300, bidirectional=True)
  (hidden2emit_score): Linear(in_features=600, out_features=5, bias=True)
)
```

o

```
{'I-Process': 6, 'I-Material': 5, 'I-Task': 7, 'B-Process': 4, 'B-Task': 3, 'B-Material': 2, '<PAD>': 1, 'O': 8, '<UNK>': 0}
```

	precision	recall	f1-score	support
2	0.00	0.33	0.01	6
3	0.00	0.00	0.00	0
4	0.09	0.19	0.12	378
5	0.01	0.23	0.02	31
6	0.23	0.23	0.23	1228
7	0.00	0.00	0.00	0
8	0.96	0.83	0.89	20388
accuracy			0.79	22031
macro avg	0.19	0.26	0.18	22031
weighted avg	0.91	0.79	0.84	22031

- Bilstm with CRF without glove(Semval 2010)
    - Architecture

```
BiLSTMCRF(
  (sent_vocab): Vocab()
  (tag_vocab): Vocab()
  (embedding): Embedding(3937, 300)
  (dropout): Dropout(p=0.5, inplace=False)
  (encoder): LSTM(300, 300, bidirectional=True)
  (hidden2emit_score): Linear(in_features=600, out_features=5, bias=True)
)
```

	precision	recall	f1-score	support
2	0.23	0.33	0.27	506
3	0.18	0.30	0.23	610
4	0.97	0.94	0.95	19553
accuracy			0.90	20669
macro avg	0.46	0.52	0.48	20669
weighted avg	0.92	0.90	0.91	20669

- Bilstm with CRF with glove(Semval 2010)
    - Architecture

```
BiLSTMCRF(
  (sent_vocab): Vocab()
  (tag_vocab): Vocab()
  (embedding): Embedding(3937, 300)
  (dropout): Dropout(p=0.5, inplace=False)
  (encoder): LSTM(300, 300, bidirectional=True)
  (hidden2emit_score): Linear(in_features=600, out_features=5, bias=True)
)
```

```
{'B-KEY': 3, 'I-KEY': 2, '<PAD>': 1, 'O': 4, '<UNK>': 0}
```

	precision	recall	f1-score	support
2	0.07	0.48	0.12	104
3	0.05	0.47	0.09	105
4	1.00	0.92	0.96	20460
accuracy			0.92	20669
macro avg	0.37	0.62	0.39	20669
weighted avg	0.99	0.92	0.95	20669

- Bilstm with CRF without glove(Inspec)
    - Architecture

```
BiLSTMCRF(
  (sent_vocab): Vocab()
  (tag_vocab): Vocab()
  (embedding): Embedding(3937, 300)
  (dropout): Dropout(p=0.5, inplace=False)
  (encoder): LSTM(300, 300, bidirectional=True)
  (hidden2emit_score): Linear(in_features=600, out_features=5, bias=True)
)
```

```
{'I-KEY': 4, 'B-KEY': 2, '<PAD>': 1, 'O': 3, '<UNK>': 0}
```

	precision	recall	f1-score	support
2	0.29	0.56	0.38	2532
3	0.95	0.90	0.92	60313
4	0.44	0.56	0.49	4447
accuracy			0.86	67292
macro avg	0.56	0.67	0.60	67292
weighted avg	0.89	0.86	0.88	67292

- Bilstm with CRF with glove(Inspec)
    - Architecture

```
BiLSTMCRF(
  (sent_vocab): Vocab()
  (tag_vocab): Vocab()
  (embedding): Embedding(3937, 300)
  (dropout): Dropout(p=0.5, inplace=False)
  (encoder): LSTM(300, 300, bidirectional=True)
  (hidden2emit_score): Linear(in_features=600, out_features=5, bias=True)
)
```

```
{'B-KEY': 3, 'I-KEY': 2, '<PAD>': 1, 'O': 4, '<UNK>': 0}
```

	precision	recall	f1-score	support
2	0.49	0.55	0.52	5042
3	0.34	0.54	0.42	3034
4	0.94	0.90	0.92	59216
accuracy			0.86	67292
macro avg	0.59	0.66	0.62	67292
weighted avg	0.88	0.86	0.87	67292

Pretrained Models

- Keybert with keyphrase-vectorizers:
  KeyphraseVectorizers package together with KeyBERT. The KeyphraseVectorizers package extracts keyphrases with part-of-speech patterns from a collection of text documents and converts them into a document-keyphrase matrix. A document-keyphrase matrix is a mathematical matrix that describes the frequency of keyphrases that occur in a collection of documents

	precision	recall	f1-score	support
B	0.43	0.17	0.25	4207
I	0.56	0.21	0.31	4875
O	0.89	0.97	0.93	58218
accuracy			0.87	67300
macro avg	0.63	0.45	0.50	67300
weighted avg	0.84	0.87	0.84	67300

- Keyphrase-generation-keybart: This model uses KeyBART as its base model and fine-tunes it on the Inspec dataset. KeyBART focuses on learning a better representation of keyphrases in a generative setting. It produces the keyphrases associated with the input document from a corrupted input. The input is changed by token masking, keyphrase masking and keyphrase replacement

	precision	recall	f1-score	support
B	0.56	0.38	0.45	4207
I	0.63	0.41	0.50	4875
O	0.91	0.96	0.94	58218
accuracy			0.88	67300
macro avg	0.70	0.58	0.63	67300
weighted avg	0.87	0.88	0.87	67300

- Keyphrase-extraction-kbir : This model uses KBIR as its base model and fine-tunes it on the Inspec dataset. KBIR or Keyphrase Boundary Infilling with Replacement is a pre-trained model which utilizes a multi-task learning setup for optimizing a combined loss of Masked Language Modeling (MLM), Keyphrase Boundary Infilling (KBI) and Keyphrase Replacement Classification (KRC).

	precision	recall	f1-score	support
B	0.63	0.65	0.64	4207
I	0.70	0.71	0.70	4875
O	0.95	0.95	0.95	58218
accuracy			0.91	67300
macro avg	0.76	0.77	0.76	67300
weighted avg	0.91	0.91	0.91	67300

- Keyphrase-extraction-distilbert: This model uses distilbert as its base model and fine-tunes it on the Inspec dataset.

	precision	recall	f1-score	support
B	0.51	0.68	0.58	4207
I	0.62	0.70	0.66	4875
O	0.96	0.92	0.94	58218
accuracy			0.89	67300
macro avg	0.70	0.77	0.73	67300
weighted avg	0.90	0.89	0.90	67300

- Keyphrase-generation-t5-small: This model uses T5-small model as its base model and fine-tunes it on the Inspec dataset. Keyphrase generation transformers are fine-tuned as a text-to-text generation problem where the keyphrases are generated.

	precision	recall	f1-score	support
B	0.40	0.34	0.37	4207
I	0.62	0.36	0.46	4875
O	0.91	0.95	0.93	58218
accuracy			0.87	67300
macro avg	0.64	0.55	0.58	67300
weighted avg	0.85	0.87	0.86	67300

Failed Approches

Neural Models

- Bilstm with Self Attention(Overfitting)-This model uses Multiheaded Attention Layer with positional encoding

```
attbilstm(
  (positionalencoding): PositionalEmbedding()
  (embedding): Embedding(13832, 256)
  (attention): MultiHeadAttention(
    (query_matrix): Linear(in_features=64, out_features=64, bias=False)
    (key_matrix): Linear(in_features=64, out_features=64, bias=False)
    (value_matrix): Linear(in_features=64, out_features=64, bias=False)
    (out): Linear(in_features=512, out_features=512, bias=True)
  )
  (encoder): LSTM(256, 256, num_layers=2, dropout=0.5, bidirectional=True)
  (fc): Linear(in_features=512, out_features=3, bias=True)
  (dropout): Dropout(p=0.5, inplace=False)
)
```

	precision	recall	f1-score	support
0	0.87	0.95	0.91	123449
1	0.16	0.00	0.01	9633
2	0.07	0.06	0.06	8426
accuracy			0.84	141508
macro avg	0.37	0.34	0.33	141508
weighted avg	0.78	0.84	0.80	141508

- Bilstm with Attention(Overfitting)- This model is a seq2seq model with Attention

```
Encoder(
  (embedding): Embedding(13832, 128)
  (lstm): LSTM(128, 128, bidirectional=True)
  (lstm_2): LSTM(384, 128, bidirectional=True)
  (out): Linear(in_features=256, out_features=3, bias=True)
)
```

	precision	recall	f1-score	support
0	0.87	1.00	0.93	123449
1	0.00	0.00	0.00	9633
2	0.00	0.00	0.00	8426
accuracy			0.87	141508
macro avg	0.29	0.33	0.31	141508
weighted avg	0.76	0.87	0.81	141508

- Failed to use contextualised embeddings with BiLSTM CRF.

Error Analysis

- Due to the overabundance of outside the keyphrase words. Words with O label. The LSTM models overfit on it and produce the same output for almost all inputs.
- Challenges in changing the tokens according to the words when using BERT encodings led to the failure of the BiLSTM CRF with contextualised embeddings.