CENG 463 - Introduction to Natural Language Processing
Fall 2023-2024
Programming Assignment 2

KARADAYI, Meriç
e2448553

January 14, 2024

1 Dependency Parsing

1.1 Implementation Details

As also task mentions I have used spacy library. After a little documentation research, I found out that the CLI documentations of spacy is much more detailed than the python file documentations. Therefore I decided to use the command line functionalities spacy provides. My implementation can work properly when spacy and spacy-transformers are installed on the environment. My implementation consists of 3 main commands.

- Preparing datasets
- Creating configuration file
- Training model

1.1.1 Preparing datasets

The training, validation and testing datasets are provided in .conllu formats. Spacy does not support files in conllu format directly. It takes data in.spacy file format. However, it is providing a function that converts the conllu file into spacy file directly. Moreover, I have divided my sentences into documents with 10 sentences as spacy suggesting.

```
#Example command
python -m spacy convert tr_imst-ud-train.conllu . --converter conllu --n-sents 10
```

1.1.2 Creating configuration file

Spacy wants a config file which is storing all settings and hyperparameters that model is going to use while training. The components which are going to used and trained in pipeline also is added on this part. I have preferred to use transformers rather than tok2vec embedders. Even though tok2vec would work more efficiently, I thought that transformers would outperform the tok2vec in the result, so I have used transformers. As I said pipeline components are being specified in the config file. Since we are implementing a dependency parser, I have only added a parser to my pipeline. After these adjustment are added as flag, spacy automatically generates a base cfg file.

```
python -m spacy init config config.cfg --lang tr --pipeline parser
--optimize accuracy --gpu --force
```

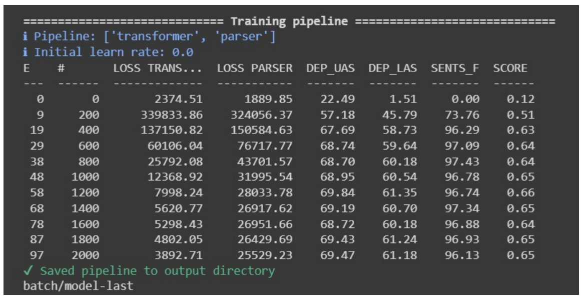
1.1.3 Training model

Base configuration file is created but some values inside the cfg file still need to be changed such as train and validation set' paths. These parameters can also be passed on this part. Furthermore all the settings in config file can be modified while calling the train command. Therefore, I have changed some parameters such batch size and number of max epoch. The default value of batch size is 128 but we do not have a too large dataset, so I think use of batches with 128 data is not going to help as we wanted. I modified it to 32. Spacy does not limit the number of epochs when it is training. There are some methods to stop like early stopping etc. but I observed after 100 epochs the learning of the model is starts to slow down.

```
python -m spacy train config.cfg --output . --gpu-id 0 -- --paths.train tr_imst-ud-train.spacy
--paths.dev tr_imst-ud-dev.spacy --training.max_epochs 100 --nlp.batch_size 32
```

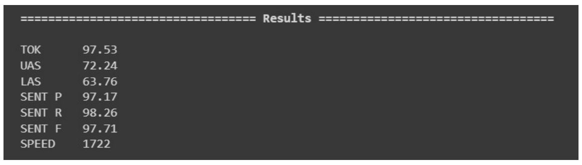
1.2 Evaluation

The benchmark of the model are as figures below.



```
=========================== Training pipeline ===========================
ℹ Pipeline: ['transformer', 'parser']
ℹ Initial learn rate: 0.0
E    #       LOSS TRANS...  LOSS PARSER  DEP_UAS  DEP_LAS  SENTS_F  SCORE
---  ------  -------------  -----------  -------  -------  -------  ------
  0       0        2374.51      1889.85    22.49     1.51     0.00    0.12
  9     200      339833.86    324056.37    57.18    45.79    73.76    0.51
 19     400      137150.82    150584.63    67.69    58.73    96.29    0.63
 29     600       60106.04     76717.77    68.74    59.64    97.09    0.64
 38     800       25792.08     43701.57    68.70    60.18    97.43    0.64
 48    1000       12368.92     31995.54    68.95    60.54    96.78    0.65
 58    1200        7998.24     28033.78    69.84    61.35    96.74    0.66
 68    1400        5620.77     26917.62    69.19    60.70    97.34    0.65
 78    1600        5298.43     26951.66    68.72    60.18    96.88    0.64
 87    1800        4802.05     26429.69    69.43    61.24    96.93    0.65
 97    2000        3892.71     25529.23    69.47    61.18    96.13    0.65
✓ Saved pipeline to output directory
batch/model-last
```

Figure 1: Evaluation metrics during training on validation set



```
================================ Results ================================

TOK      97.53
UAS      72.24
LAS      63.76
SENT P   97.17
SENT R   98.26
SENT F   97.71
SPEED    1722
```

Figure 2: Evaluation metrics on testing

- Unlabeled attachment score: 0.7224
- Labeled attachment score: 0.6376

Labeled attachment score (LAS) and unlabeled attachment score (UAS) are both metrics used to evaluate the performance of dependency parsers. They analyze the grammatical structure of sentences by determining the relationships between words. Labeled Attachment Score evaluates both the correct attachment of words and the correct identification of the relationships between the words. However Unlabeled Attachment Score focuses only on the correct attachment of words. It does not consider the relationships between words.

$$\text{LAS} = \frac{\text{\# of Correct Attachments and Labels}}{\text{Total Number of Words}}$$

$$\text{UAS} = \frac{\text{\# of Correct Attachments}}{\text{Total Number of Words}}$$

UAS is approximately 10 percent more than the LAS. When the formulas is considered, it makes totally sense. Because the correct predictions LAS is marking is basically a subset of the correct prediction UAS is marking. LAS does not count the words whose attachment is correct but label is incorrect unlike UAS does.

1.3 Experiment

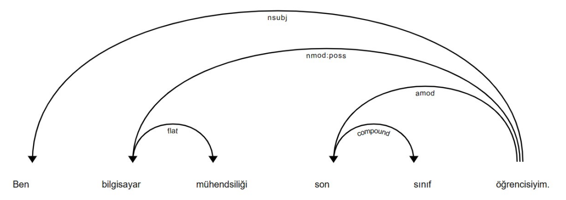


Figure 3: Correct parsing

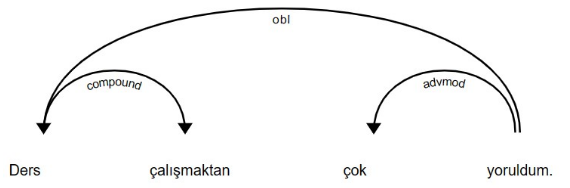


Figure 4: Correct parsing

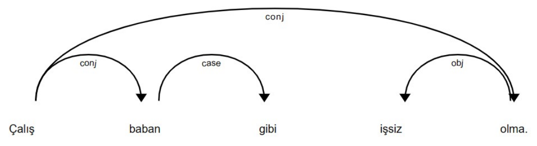


Figure 5: Incorrect parsing

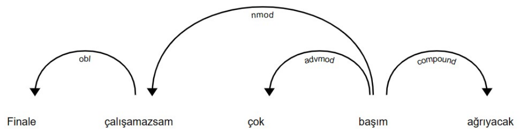


Figure 6: Incorrect parsing

The correct parsing results are sentences which are grammatically totally correct, so it performed a better result. However, turkish is a flexible language and words on sentences can be in inverse order, they do not need to have a strict order, so this caused our model to fail. In addition soma sentence can be derived to multiple meanings which people can understand from the context, this also a weak side which cause false predictions.

2 Extractive Summarization

2.1 Implementation Details

My implementation consist of phases.

- Take file name as input
- Create vectorizer
- Getting most informative sentences

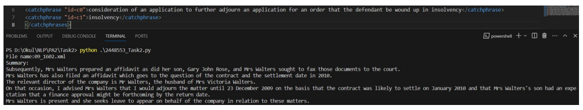
2.1.1 Creating vectorizer

1. Reads all the file's name ending with ".xml"
2. Parses XML files and extract the sentences from each file
3. Creates a mega document which each element is merge of sentence of each cases
4. Create a IDF vectorizer according to megadoc.

2.1.2 Getting most informative sentences

1. Create a TF-IDF matrix specific the file user entered using the created vectorizer
2. Calculate cos similarities and return the 5 sentences which has the largest cos similarity value on average. These are the most informative sentences

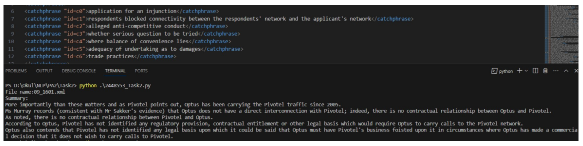
2.2 Experiment



```
<catchphrase "id=c0">consideration of an application to further adjourn an application for an order that the defendant be wound up in insolvency</catchphrase>
<catchphrase "id=c1">insolvency</catchphrase>
</catchphrases>
```

```
PS D:\Okul\NLP\PA2\Task2> python .\2448553_Task2.py
File name:09_1602.xml
Summary:
Subsequently, Mrs Walters prepared an affidavit as did her son, Gary John Rose, and Mrs Walters sought to fax those documents to the court.
Mrs Walters has also filed an affidavit which goes to the question of the contract and the settlement date in 2010.
The relevant director of the company is Mr Walters, the husband of Mrs Victoria Walters.
On that occasion, I advised Mrs Walters that I would adjourn the matter until 23 December 2009 on the basis that the contract was likely to settle on January 2010 and that Mrs Walters's son had an expe
ctation that a finance approval might be forthcoming by the return date.
Mrs Walters is present and she seeks leave to appear on behalf of the company in relation to these matters.
```

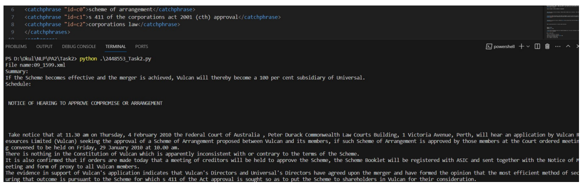
Figure 7: Experiment-1



```
<catchphrase "id=c0">application for an injunction</catchphrase>
<catchphrase "id=c1">respondents blocked connectivity between the respondents' network and the applicant's network</catchphrase>
<catchphrase "id=c2">alleged anti-competitive conduct</catchphrase>
<catchphrase "id=c3">whether serious question to be tried</catchphrase>
<catchphrase "id=c4">where balance of convenience lies</catchphrase>
<catchphrase "id=c5">adequacy of undertaking as to damages</catchphrase>
<catchphrase "id=c6">trade practices</catchphrase>
```

```
PS D:\Okul\NLP\PA2\Task2> python .\2448553_Task2.py
File name:09_1601.xml
Summary:
More importantly than these matters and as Pivotel points out, Optus has been carrying the Pivotel traffic since 2005.
Ms Murray records (consistent with Mr Sakker's evidence) that Optus does not have a direct interconnection with Pivotel; indeed, there is no contractual relationship between Optus and Pivotel.
As noted, there is no contractual relationship between Pivotel and Optus.
According to Optus, Pivotel has not identified any regulatory provision, contractual entitlement or other legal basis which would require Optus to carry calls to the Pivotel network.
Optus also contends that Pivotel has not identified any legal basis upon which it could be said that Optus must have Pivotel's business foisted upon it in circumstances where Optus has made a commercia
l decision that it does not wish to carry calls to Pivotel.
```

Figure 8: Experiment-2



```
<catchphrase "id=c0">scheme of arrangement</catchphrase>
<catchphrase "id=c1">s 411 of the corporations act 2001 (cth) approval</catchphrase>
<catchphrase "id=c2">corporations law</catchphrase>
</catchphrases>
```

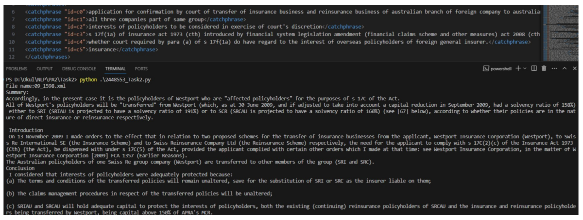
```
PS D:\Okul\NLP\PA2\Task2> python .\2448553_Task2.py
File name:09_1599.xml
Summary:
If the Scheme becomes effective and the merger is achieved, Vulcan will thereby become a 100 per cent subsidiary of Universal.
Schedule:


 NOTICE OF HEARING TO APPROVE COMPROMISE OR ARRANGEMENT



 Take notice that at 11.30 am on Thursday, 4 February 2010 the Federal Court of Australia , Peter Durack Commonwealth Law Courts Building, 1 Victoria Avenue, Perth, will hear an application by Vulcan R
esources Limited (Vulcan) seeking the approval of a Scheme of Arrangement proposed between Vulcan and its members, if such Scheme of Arrangement is approved by those members at the Court ordered meetin
g convened to be held on Friday, 29 January 2010 at 10.00 am.
There is nothing in the Constitution of Vulcan which is apparently inconsistent with or contrary to the terms of the Scheme.
It is also confirmed that if orders are made today that a meeting of creditors will be held to approve the Scheme, the Scheme Booklet will be registered with ASIC and sent together with the Notice of M
eeting and form of proxy to all Vulcan members.
The evidence in support of Vulcan's application indicates that Vulcan's Directors and Universal's Directors have agreed upon the merger and have formed the opinion that the most efficient method of sec
uring that outcome is pursuant to the Scheme for which s 411 of the Act approval is sought so as to put the Scheme to shareholders in Vulcan for their consideration.
```

Figure 9: Experiment-3



```
<catchphrase "id=c0">application for confirmation by court of transfer of insurance business and reinsurance business of australian branch of foreign company to australia
<catchphrase "id=c1">all three companies part of same group</catchphrase>
<catchphrase "id=c2">interests of policyholders to be considered in exercise of court's discretion</catchphrase>
<catchphrase "id=c3">s 17f(1a) of insurance act 1973 (cth) introduced by financial system legislation amendment (financial claims scheme and other measures) act 2008 (cth
<catchphrase "id=c4">whether court required by para (a) of s 17f(1a) do have regard to the interest of overseas policyholders of foreign general insurer.</catchphrase>
<catchphrase "id=c5">insurance</catchphrase>
</catchphrases>
```

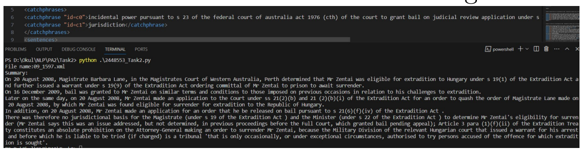
```
PS D:\Okul\NLP\PA2\Task2> python .\2448553_Task2.py
File name:09_1598.xml
Summary:
Accordingly, in the present case it is the policyholders of Westport who are "affected policyholders" for the purposes of s 17C of the Act.
All of Westport's policyholders will be "transferred" from Westport (which, as at 30 June 2009, and if adjusted to take into account a capital reduction in September 2009, had a solvency ratio of 158%)
 either to SRI (SRIAU is projected to have a solvency ratio of 191%) or to SCR (SRCAU is projected to have a solvency ratio of 160%) (see [67] below), according to whether their policies are in the nat
ure of direct insurance or reinsurance respectively.

 Introduction
 On 13 November 2009 I made orders to the effect that in relation to two proposed schemes for the transfer of insurance businesses from the applicant, Westport Insurance Corporation (Westport), to Swis
s Re International SE (the Insurance Scheme) and to Swiss Reinsurance Company Ltd (the Reinsurance Scheme) respectively, the need for the applicant to comply with s 17C(2)(c) of the Insurance Act 1973
(Cth) (the Act), be dispensed with under s 17C(5) of the Act, provided the applicant complied with certain other orders which I made at that time: see Westport Insurance Corporation, in the matter of W
estport Insurance Corporation [2009] FCA 1357 (Earlier Reasons).
The Australian policyholders of one Swiss Re group company (Westport) are transferred to other members of the group (SRI and SRC).
Conclusion
 I considered that interests of policyholders were adequately protected because:
(a) The terms and conditions of the transferred policies will remain unaltered, save for the substitution of SRI or SRC as the insurer liable on them;

(b) The claims management procedures in respect of the transferred policies will be unaltered;

(c) SRIAU and SRCAU will hold adequate capital to protect the interests of policyholders, both the existing (continuing) reinsurance policyholders of SRCAU and the insurance and reinsurance policyholde
rs being transferred by Westport, being capital above 150% of APRA's MCR.
```

Figure 10: Experiment-4



```
<catchphrases>
<catchphrase "id=c0">incidental power pursuant to s 23 of the federal court of australia act 1976 (cth) of the court to grant bail on judicial review application under s
<catchphrase "id=c1">jurisdiction</catchphrase>
</catchphrases>
<sentences>
```

```
PS D:\Okul\NLP\PA2\Task2> python .\2448553_Task2.py
File name:09_1597.xml
Summary:
On 20 August 2008, Magistrate Barbara Lane, in the Magistrates Court of Western Australia, Perth determined that Mr Zentai was eligible for extradition to Hungary under s 19(1) of the Extradition Act a
nd further issued a warrant under s 19(9) of the Extradition Act ordering committal of Mr Zentai to prison to await surrender.
On 16 December 2009, bail was granted to Mr Zentai on similar terms and conditions to those imposed on previous occasions in relation to his challenges to extradition.
Later on the same day, on 20 August 2008, Mr Zentai made an application to the Court under ss 21(2)(b) and 21 (2)(b)(i) of the Extradition Act for an order to quash the order of Magistrate Lane made on
 20 August 2008, by which Mr Zentai was found eligible for surrender for extradition to the Republic of Hungary.
In addition, on 20 August 2008, Mr Zentai made an application for an order that he be released on bail pursuant to s 21(6)(f)(iv) of the Extradition Act .
There was therefore no jurisdictional basis for the Magistrate (under s 19 of the Extradition Act ) and the Minister (under s 22 of the Extradition Act ) to determine Mr Zentai's eligibility for surren
der (Mr Zentai says this was an issue addressed, but not determined, in previous proceedings before the Full Court, which granted bail pending appeal); Article 3 para (1)(f)(ii) of the Extradition Trea
ty constitutes an absolute prohibition on the Attorney-General making an order to surrender Mr Zentai, because the Military Division of the relevant Hungarian court that issued a warrant for his arrest
 and before which he is liable to be tried (if charged) is a tribunal 'that is only occasionally, or under exceptional circumstances, authorised to try persons accused of the offence for which extradit
ion is sought'.
```

Figure 11: Experiment-5

2.3 Enhancement

We have used TF-IDF embedding for this task, however instead of relying on TF-IDF, consider using more advanced sentence embedding techniques can provide better performance. For example, we have used transformers on Task-1, if we have used transformers also in this this would help our model to perform better. Furthermore we did not add any semantic relationship or meanings to our words during training, adding such features can enhance the performance. We have compared our sentences with cosine similarities but we did not perform any clustering. If we use clustering algorithms, we could provide better summary option with our model.

2.4 Evaluation

According to my research on how to evaluate a summarization algorithm. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) which is a set of metricss can be used primarily is such tasks. The main point of ROUGE is that it measures overlapping n-grams. There is a library called rouge in Python. That libraries is evaluating the generated summary with respect to a reference summary. The case files have also catchphrase information, we could use merge this catchphrases and take it as reference summary to evaluate our model with rogue.