DIVE INTO NLP ANALYSIS

COMPARISON OF DIFFERENT APPROACHES TO NERC, SENTIMENT AND TOPIC ANALYSIS

INTRODUCTION

In the modern world the vast majority of data is unstructured and therefore inaccessible for machines. Around 80% of all the information stored in the internet has to be processed in order for it to become useful in providing knowledge. An area that is specialized in making that happen is called NLP - Natural Language Processing. In the era of big data it has become indispensable for extracting meaning and insight from different kinds of text. This applies to not only professional texts like articles or scientific papers, but as well social media analytics and in general understanding how people communicate and what do they talk about. In this paper we take a look at three different areas of NLP, Named Entity Recognition and Classification (NERC), Sentiment Analysis and Topic Analysis. The first one focuses on recognizing and later classifying potential key entities in a given text, assigning them a label which for example could be people, locations or events. Sentiment analysis specializes in assessing a sentiment of a given sentence or text, determining whether it has a positive, neutral or negative tone. Finally topic analysis is tasked with discovering topic of a given sentence, shorter text or a longer text. For the comparison within each of the areas of NLP we focused on in this research we took a similar approach. Relying on literature we determined what is the most common method used in the field, to set it as a benchmark for performance and compared it with a model based on a Bidirectional encoder model - BERT. For NERC we juxtaposed a machine learning model - Support Vector Machine (SVM) and bert-base-ner, which is a pre-trained BERT model fine tuned for NER tasks. In sentiment analysis we have opted for comparing a rule based approach, by using a SentiWordNet lexicon and a deep learning approach in the form of Twitter-roBERTa-base model, specialized for sentiment analysis of twitter posts. Lastly for topic analysis we set side by side a Latent Dirichlet Allocation (LDA) model , which is a generative probabilistic model for data collection, and transformer based BERTopic model, which using sentence embedding techniques predicts the topic. In each segment of this research we aim to thoroughly analyze and compare each pair of methods in terms of their performance in various aspects, derive where the differences come from, while keeping in mind several limitations we have to consider conducting these experiments. While each method has its own merits and trade-offs, placing them side by side within the same experimental framework allows us to identify not only which approach tends to excel in a given context but also which might adapt better to new challenges or evolving data conditions.

DIVISION OF WORK

Jan Burakowski
- Sentiment Analysis Code
- Sentiment Analysis description
- Sentiment Analysis Poster Preparation

Mateusz Kielan
- Topic Analysis Code
- Topic Analysis description
- Topic Analysis Poster Preparation

Oliwier Augustynowicz
- Sentiment Analysis Code
- Introduction writing
- Introduction and Sentiment Analysis Poster Preparation

Szymon Czternasty:
- NERC Code
- NERC Analysis
- NERC Description and Poster Preparation

NAMED ENTITY RECOGNICTION AND CLASSIFICATION

Objective

The goal of Named Entity Recognition (NERC) is to automatically identify and classify words or phrases in a text into predefined entity categories, such as persons, locations, organizations, and miscellaneous entities. NERC is a fundamental task in Natural Language Processing (NLP) and plays a crucial role in NLP pipeline. By assigning appropriate entity labels to words, NERC helps structure unstructured text data, making it more useful for further analysis.

Methods

To accomplish the Named Entity Recognition (NERC) task, three different models were implemented and evaluated. The models differ in their feature representations and learning mechanisms, allowing for a comparative analysis of traditional machine learning approaches versus deep learning-based methods.

SVM (POS & Token Features, No Embeddings)

A Support Vector Machine (SVM) classifier was trained using only token-based and part-of-speech (POS) features. This model follows a traditional machine learning approach, where manually engineered features play a key role in entity classification. Each token was represented by its surface form and its corresponding POS tag, which provides syntactic context. The advantage of such a model is that it is computationally efficient and interpretable. However it lacks semantic understanding, struggles with polysemous words and context-dependent entity recognition.

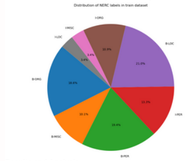
SVM (GoogleNews-vectors-negative300 Word Embeddings)

To improve upon the baseline SVM model, pre-trained word embeddings from Google's Word2Vec model (GoogleNews-vectors-negative300) were integrated as features. These embeddings capture semantic relationships between words by mapping them into a continuous vector space. Each token was represented using its 300-dimensional word embedding from Word2Vec, along with POS tags.Word embeddings provide semantic generalization, helping the model recognize similar entities. Nevertheless, the polysemous flaw of the previous model is not identified due to the fact that Word2Vec embeddings are static, meaning they do not account for word meaning changes based on context

Fine-Tuned Transformer Model (dslim/bert-base-NER)

Finally, the pre-trained BERT model (dslim/bert-base-NER) fine-tuned on NERC task was used. Studies[1][2] show that pre-trained, fine-tuned BERT models yield impressive results when it comes to NERC tasks. The model was pre-trained on the same data as SVM to ensure the similar setup for the comparison. Unlike SVM models, BERT leverages deep contextualized embeddings by considering the entire sentence when making predictions. Tokens are embedded using BERT's transformer-based architecture, capturing both local and global contextual dependencies. The significant benefit of using such a model is strong generalization, robust entity recognition, and ability to disambiguate entity types using contextual information. Though, it comes at a cost of computationally expensiveness.

Models were evaluated using the same systematic approach on the provided test set, by calculating precision, recall, F1-score, and overall accuracy along with micro and macro averages.

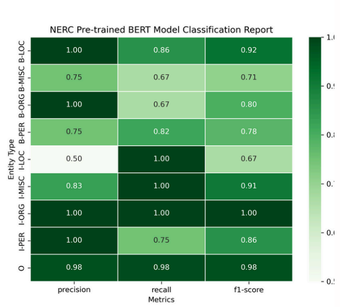


Data

In order to ensure the solid, reproducible performance, robust dataset had to be used with human annotated labels. For this assignment, models were trained on the CoNLL-2003 dataset, a widely used benchmark containing newswire text annotated with four entity types: PER (person), ORG (organization), LOC (location), and MISC (miscellaneous), containing roughly 200 000 examples evenly distributed across different. Since the test set contained some more detailed entities types we decided to classify all others as a miscellaneous for consistency, this is also due to the fact that the BERT model we used was already pre-trained on the those 4 categories of entities. Why did we choose CoNLL-2003? Well because it became an industry leading dataset, being used extensively in NLP research, making it a gold standard for evaluating NERC models. Additionally, Many state-of-the-art models, including BiLSTM-CRF (Lample et al., 2016)[3] and Transformer-based approaches (Devlin et al., 2019)[2], were first validated on this dataset. The dataset includes a balanced mix of entity types (e.g., locations, organizations, persons). Unlike domain-specific datasets, CoNLL-2003 generalizes well across different textual genres. The dataset includes full news articles, allowing models to leverage contextual dependencies rather than just isolated sentence-level NER. This is particularly beneficial for transformer-based models like BERT, which rely on context to distinguish between ambiguous entities.

Results and Analysis

The three models tested for Named Entity Recognition (NERC) exhibited distinct strengths and weaknesses in their classification performance.



The first model, the SVM trained with only token-based and part-of-speech (POS) features, achieved an overall accuracy of 85% with a macro F1-score of 0.33. While it performed well on frequently occurring entity types such as B-LOC with an F1-score of 0.62, it struggled with many other entity classes. The recall for I-PER was relatively high at 0.88, indicating that the model was somewhat effective at capturing these mentions once identified. However, it failed entirely on categories such as I-LOC, I-MISC, and I-ORG which indicate that the model struggles with multi-word entities. This suggests that the model's reliance on simple token and POS features was insufficient for generalizing across all entity types.

The second model, the SVM using pre-trained GoogleNews Word2Vec embeddings, achieved the same overall accuracy of 85% but with a higher macro F1-score of 0.41. This model showed a significant improvement in B-PER, where the F1-score increased from 0.53 to 0.61, and I-PER, where the F1-score increased from 0.42 to 0.50. The most notable improvement was in B-ORG, where recall increased to 1.00, though precision was low at 0.23, indicating that while the model was able to capture all occurrences of B-ORG, it also introduced a high number of false positives. Despite these improvements, the model still failed to classify B-MISC and I-MISC.. The results suggest that adding word embeddings helped improve recall and generalization to some extent, but the model still struggled with rare entity types and displayed a tendency to misclassify certain categories

The two SVM models tend to make the same mistake by having trouble with names of places. The example errors are Manchester classified as Location when in fact its part of the sport team name Manchester United (ORG) or Bristol identified as organization when it should have been classified as a location. The other errors are long names consisting of several elements like "To kill a Mockingbird" classified as O or missing parts of some entities like "Queen Elizabeth the II" with only Elizabeth identified correctly as a person. Those errors stems from the fact that those models lack context awareness..

The third model, a fine-tuned transformer using the BERT model, achieved a significantly higher overall accuracy of 95% and a macro F1-score of 0.85. This model outperformed both SVM models across all metrics, demonstrating high precision and recall across all entity types. The F1-scores for B-LOC, B-MISC, and B-PER were 0.92, 0.71, and 0.78, respectively, showing a well-balanced performance. Unlike the SVM models, this model did not entirely fail on any entity class, and it demonstrated the ability to recognize even rare entities with reasonable accuracy. While the recall for some categories such as B-ORG was slightly lower than its precision, the overall results indicate strong generalization, which is expected given the contextual embeddings learned by BERT. The model made mistakes in the reasonable cases like confusing Oppenheimer and Barbie as Person instead of Movie name. There are a couple cases when the Beginning and Inner class are confused. This shows that even if context awareness play crucial role, even BERT makes mistakes sometimes.

	token	predicted	gold
0	Coldplay	B-MISC	B-ORG
3	German	B-MISC	O
5	Barbie	B-PER	B-MISC
6	Oppenheimer	B-PER	B-MISC
10	Elizabeth	B-PER	I-PER
1	Stadium	I-LOC	B-LOC
7	a	I-MISC	O
8	a	I-MISC	O
2	Banksy	O	B-PER
4	1984	O	B-MISC

In summary, the fine-tuned BERT model was the most effective, significantly outperforming the SVM models. The traditional SVM approach using token and POS features struggled with rare entities, while the addition of pre-trained word embeddings improved recall but did not fully address class imbalances. The transformer-based model demonstrated why deep contextualized embeddings are crucial for high-quality Named Entity Recognition, confirming findings from prior research[1][2][3] that transformers consistently outperform classical machine learning models in sequence labeling tasks.

Limitation

One major limitation of this experiment is the focus on a limited set of basic entity types, primarily persons, locations, and organizations, while neglecting more complex or domain-specific categories. The CoNLL-2003 dataset, used for training, is well-suited for general-purpose Named Entity Recognition (NERC) but lacks coverage of fine-grained entities like work of art, or even domain-specific entities. As a result, the models may not generalize well to other domains.

Moreover, the experiment only evaluates models on a single dataset, limiting the ability to generalize findings to other kinds of text with different linguistic styles, languages, terminologies, or entity distributions.

SENTIMENT ANALYSIS

Objective

Our objective was to analyse and compare the performance of the rule-based approach using the SentiWordNet lexicon and the Deep Learning approach using a pre-trained version of the Bidirectional Encoder model BERT specialised in Twitter posts, as the structure of test data resembles the format that can be noticed in Twitter posts.

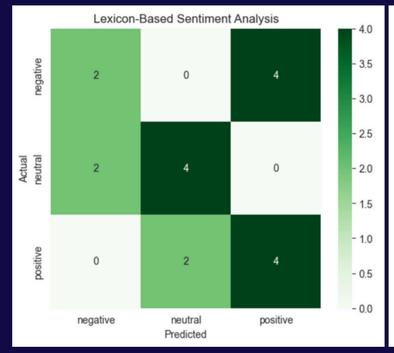
We have chosen the SentiWordNet lexicon for the rule-based approach, as it has annotated sentiment labels for the whole WordNet lexicon [1], making it very generalizable across different applications of sentiment analysis in English and in this approach we wanted to focus on the most basic generalizable approach to see how it compares to specialized one.

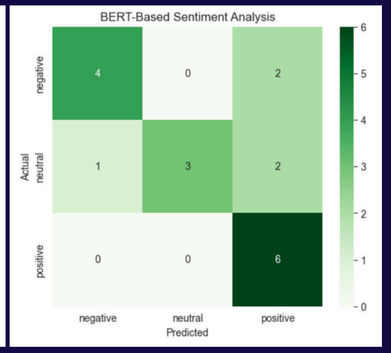
For the Deep Learning approach, we have chosen to use the Twitter-roBERTa-base for Sentiment Analysis model. This choice was motivated by the findings of the TWEETEVAL benchmark [2], which demonstrates that pre-trained transformer models adapted specifically to Twitter data significantly outperform general-purpose models on tweet classification tasks, including sentiment analysis.

Methods

In the case of SentiWordNet lexicon sentiment classification, we had to perform text pre-processing so that the test data would match the appropriate format. For this task, we have used the SpaCy module for tokenization, lemmatization, and Part-of-Speech tagging. For sentiment classification, we used the sentiwordnet function from the NLTK toolkit and wrote a function that assigns the sentiwordnet score to each token in a sentence and calculates the final verdict.

For the Deep Learning approach, the task was a bit simpler, as the pre-trained model is a part of the transformers library, so we just created a pipeline for sentiment classification where we have chosen the "cardiffnlp/twitter-roberta-base-sentiment" model and mapped the labels from numbers to classes of sentiment for readability.





Results

As you can see on confusion matrixes for both approaches, the Deep Learning approach is slightly better as it classified the sentiment correctly in 13 out of 18 cases, while the Rule-based approach classified only 10 cases correctly.

From the confusion matrix, we can see that the lexicon-based approach especially falls short in the classification of negative cases and often classifies them as positive. This is most likely because of the negation problem and the fact that only distinct tokens are graded, and in many cases, a whole context is required to understand the sentiment, like in the case of *"It's really incredibly impressive to mess up such a tested blockbuster formula."* which was falsely classified as positive, where distinct words can have high scores in the lexicon but when you connect them the whole sentence becomes negative.

The biggest flaw of the Deep-learning approach is also the misclassification of positive labels, yet in this case, the problem might be more complex, as the labels are not only misclassified as negative but also neutral. The model takes into account the context of each token but most likely struggles with detecting irony, like in the case of *"It's really incredibly impressive to mess up such a tested blockbuster formula."* which was misclassified as positive or has difficulty with the distinction of the most important sentiment if multiple are provided in a phrase like in the case of *"It's more of a slow burn than a page-turner, but it's well-written, I guess."* which is also misclassified as positive.

Overall, we can conclude that the pre-trained version of the Bidirectional Encoder model BERT, specialised in Twitter posts, performs better than the basic lexicon-based approach but still struggles in more complex cases, which include irony or multiple sentiments.

TOPIC ANALYSIS

Objective

In the topic analysis part the performance of 2 models are gonna be assessed: LDA and BERTopic. LDA provides a steady benchmark for a conventional topic modelling task. BERTopic was selected as a second model as it provides a fundamentally different approach to the task at hand. While LDA is a generative probabilistic model for data collections [7], BERTopic is a transformer based technique consisting of 2 main stage processes:

1. Embedding - utilizing a transformer model (BERT).
2. KMeans - group similar sentence embeddings.

This allows BERTopic to capture context as it utilizes transformer models while staying unsupervised in the training phase (with k-means) [8]. Therefore, comparing those models will aid in understanding whether the probabilistic approach or transformer-based model performs better on a short general topic test-set.

Data

For the topic analysis an ag_news dataset was used as training data. The dataset was selected because it contains a sufficient number of articles that have topics as well as contains both the title and text of the article itself. Furthermore, it was used in multiple studies which utilized LDA [4], embedding spaces [5] as well as in semi-supervised topic modelling (Xu et al., 2023). Further, it contains 4 largest classes: "World","Sports", "Business" and "Sci-Tech" this ensures diversity among data entry instances as well as well as related topics to the one in the test data e.g "Sports". Therefore, the ag_news dataset is suitable for performing the topic analysis with our approach. The dataset was taken from Hugging Face.

Methods

Preprocessing for LDA and BERTopic differs. For LDA tokenization,lemmatization and stemming were performed as introduced in the Lab 6 of the course. Furthermore, the stopwords removal was added as those can drastically lower the performance of LDA. The LDA was performed for both BoW representation and Tf-Idf Representation similarly to the Lab assignments. The former was created using gensim doc2bow and the later using TfidfModel.

For BERTopic no preprocessing is needed. The tokenization is done internally whereas removing Stopwords, stemming and lemmatization could impact the context which BERTopic aims to capture[9]. Therefore, data was passed without preprocessing. The model can be trained using HDBSCAN which assigns topics based on clusters and probabilities, automatically defines the number of topics and defines outliers or using K-means which acts on the set number of topics/clusters.

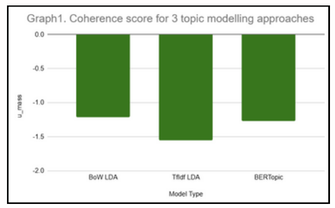
A Preliminary run of the HDBSCAN showed that because it identifies large amount of topics specific in the training data it classifies all instances of the test data as outliers as the test dataset is small and contains too few topics to mach. Therefore K-means approach was selected in order to match the number of topics in LDA (10) and make sure that every article is assigned to a topic therefore avoiding the outlier issue in the first approach.

As an evaluation for the models coherence *u_mass* score will be used. It was selected as it has a high compatibility with a short-text datasets (test-set) allowing us to provide a quantitative evaluation. Although it is known to be less aligned with the human interpretability it allows us to avoid the generation of undefined data in the metrics such ass *c_v* and *c_npmi[10]* The score was computed using gensim Coherence Model.

Results

The results are displayed on the Graph 1. The u_mass results rank from the lowest being the most optimal, to the highest score being least optimal. The results are the following. The TfIdf LDA model performed the best across 3 models with the -1.55 score indicating the best topic word co-occurrence in the text corpora. This could be due to the TfIdf being based on the word co-occurrences, which are evaluated by the coherence value afterwards. Furthermore, as it penalizes the words that are too frequent across the topics it could highlight the importance of topic-sensitive words and their co-occurrence across the articles. The second model was the BERTopic with a score of -1.27. It outperformed the classical Bag-Of-Words LDA as it utilizes the transformer models that learn context from the data and can deal better with the polysemy of words.

However, it could still perform worse than the TfIdf LDA for a couple of reasons.The u_mass score is more tailored to the tf-idf and BoW models as it evaluates the topic word co-occurrences in the text corpora. While BERTopic managed to outperform the classical BoW approach on a small test-set due to its better context reading, tf-idf better captured word co-occurrences. Further evaluation is presented in the Extension And Limitation section.

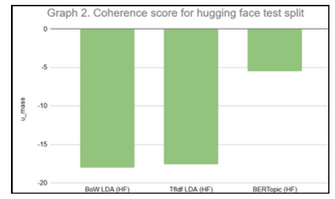


Extension And Limitation

As an extension to the evaluation of the model performance, the same coherence measure was run for the test split on the Hugging Face dataset. Because all 3 models were trained on the train split the performance is expected to be much higher.

The results were the following. The performance of all 3 models improved significantly compared to the initial test dataset due to the topics being better fitted for the hugging face test set. This could indicate 2 main points. First, a poor generalization of all 3 models for the datasets containing more general topic/language. Second, it suggests a big discrepancy between the training dataset and the test dataset as the models performed significantly better on the test splits from HF dataset. Although the training dataset contained diverse topic and language, it still wasn't enough to generalize to the provided test set.

Furthermore, the finding may suggest that the u_mass coherence score favors the models relying on co-occurrence such as Tf-Idf and BoW models. However, in order to validate the claim further research is required which is explained in the Future work section



The main limitation of the research lies in the coherence parameters. As we were unable to asses the performance with more human judgment correlated measures only one was utilized. Furhtermore, the research could incorporate other visualization such as topic density or word to topic vector representations.

Moreover, as mentioned before large discrepancies between train sets and test sets could have negatively impacted the results.

FUTURE WORK

Future work could extend this analysis by evaluating models on multiple datasets, including those with more diverse entity annotations, to assess robustness and adaptability in real-world applications. Furthermore we could consider broader evaluation metrics, by including measures for robustness, fairness and bias or some user-focused metrics. Additionally universal model capable of performing each of those 3 tasks by itself could be compared with each model that specializes in them and we would evaluate their performance gap. For topic evaluation in order to further evaluate the performance of the 3 topic models they could be evaluated using other mentioned coherence scores such as c_v or c_npmi to check if the u_mass coherence score is favouring the LDA models.

DATA SETS

- CONLL2003
- AG NEWS (HTTPS://PAPERSWITHCODE.COM/DATASET/AG-NEWS)

REFERENCES

[1] Akbik et al. (2019) – Contextual String Embeddings for Sequence Labeling

[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT.

[3] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural Architectures for Named Entity Recognition. ACL.

[4] Srivastava, A., & Sutton, C. (2017, March 4). Autoencoding variational inference for topic models. arXiv.org. https://arxiv.org/abs/1703.01488

[5] Dieng, A. B., Ruiz, F. J. R., & Blei, D. M. (2019, July 8). Topic Modeling in embedding spaces. arXiv.org. https://arxiv.org/abs/1907.04907

[6] Xu, W., Jiang, X., Rao, S. S. H., Iannacci, F., & Zhao, J. (2023). vONTSS: vMF based semi-supervised neural topic modeling with optimal transport. Findings of the Association for Computational Linguistics: ACL 2022, 4433–4457. https://doi.org/10.18653/v1/2023.findings-acl.271

[7] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of Machine Learning Research, 3, 993–1022. https://doi.org/10.5555/944919.944937

[8] Grootendorst, M. (2022, March 11). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv.org. https://arxiv.org/abs/2203.05794

[9] Grootendorst, M. P. (n.d.). Tips & tricks - BERTopic. https://maartengr.github.io/BERTopic/getting_started/tips_and_tricks/tips_and_tricks.html?

[10]Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the Space of Topic Coherence Measures. ESF. https://doi.org/10.1145/2684822.2685324