Natural Language Processing

Natural Language Processing

- Natural Language Processing is a **cross-disciplinary** research field that draws heavily from **artificial intelligence** (AI), **machine learning** (ML), mathematics, and linguistics.
- Personal assistants, recommender systems, fake news identification, financial stock analysis, chatbots, autocorrection, auto-completion, intelligent search engines, and automatic translation or captioning are just a few examples of how NLP and AI are helping us to manage the flood of data. However, systems to process natural language are far from perfect, which leaves much space for research.
- Some of the areas we work are:
  - Natural language understanding
  - Paraphrase detection
  - Text summarization
  - Media bias/Fake news detection
  - Semantic analysis/extraction
  - Sentiment analysis

For a complete list of our research topics visit our website!

Gelections - The German Elections Under the Microscope

Background

The Elections always bring heat discussions over different topics. The increasingly use of social media amplifies these discussions as topics are discussed closely between political parties, possible candidates, and general public. In 2016, Twitter played a decisive role in US presidential election and later in the UK's Brexit referendum. In 2022 (and possibly the years to come) this situation will only get stronger. We plan to bring the German elections into a microscope and apply NLP and IR techniques to better understand the political parties, their candidates and position on their plans for the country.

Goal

- Understand the stance (for/against) of political parties and their members wrt their programs in Germany.

Tasks

- What are the most discussed topics of election programs on Twitter?
- Which topics have the most/least dis-/agreement?
- How do political parties dis-/agree on election topics with their members?

Jan Philip Wahle
wahle@gipplab.org

Terry L. Ruas
ruas@gipplab.org

Plagiarism Detection with a Multi-Task Perspective

Background

The recent success in NLP can be attributed to self-supervised learning on massive text corpora. Through self-supervision, language models learn a broad set of skills and pattern recognition abilities. Also in plagiarism detection, language models have experienced great success. However, many of the required skills to solve a single task (e.g., author identification) are also present in related tasks (e.g., originality) for which labeled data exists. Therefore, a recent trend in improving language models prior to fine-tuning has become multi-task learning which leverages labeled training data to learn many skills simultaneously. As plagiarism has different forms, such as paraphrasing, idea plagiarism, author similarity, it is an intuitive candidate to perform multi-task learning.

Goal

- Explore multi-task learning for plagiarism with neural language models

Tasks

- Train detection models based on previous state-of-the-art work
- Propose training architectures and paradigms.
- Evaluate with human studies and automated metrics

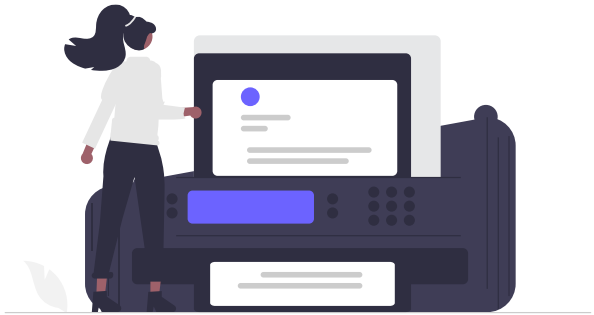
Jan Philip Wahle
wahle@gipplab.org

Terry L. Ruas
ruas@gipplab.org

Generating Paraphrased Plagiarism with GPT-3

Background

To identify neural-paraphrased plagiarism, we need data to learn which features make paraphrased examples so convincing. We ideally seek for automated solutions as they are scalable. With data paraphrased by multiple techniques, we can optimize detection methods that are robust and generalize well to unseen scenarios. We further assume generative language models paraphrase similar to humans. If we can confirm this hypothesis, and generate training data automatically, we can increase the performance of detection methods without the tedious process of finding real-world plagiarized examples.



Goal

- Explore the generation of machine-paraphrased plagiarism with neural language models.

Tasks

- Use existing datasets and find their weak spots to extend them to be more robust.
- Propose paraphrasing methods and test them using human studies.
- Evaluate whether neural-paraphrasing is close to how humans paraphrase text.

Jan Philip Wahle
wahle@gipplab.org

Terry L. Ruas
ruas@gipplab.org

Identifying Neural-Paraphrased Plagiarism

Background

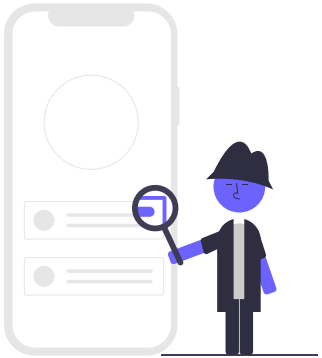
Machine-paraphrasing has become a concerning problem for research institutions, publishers, and schools, as anyone can obtain access to free tools that generate convincing plagiarism. Large auto-regressive language models with more than a hundred billion parameters, such as GPT-3, can generate text indistinguishable from human writing which makes plagiarism effortless, yet extremely difficult to spot. In the near future, when large language models become more accessible, the number of plagiarized texts increases dramatically. Therefore, we need automated plagiarism detection solutions now before models are widely misused for plagiarism

Goal

- Explore machine-paraphrased plagiarism with neural language models

Tasks

- Build detection models based on previous state-of-the-art work.
- Propose training architectures and paradigms.
- Evaluate with human studies and automated metrics.



Jan Philip Wahle
wahle@gipplab.org

Terry L. Ruas
ruas@gipplab.org

Multi-Source Meeting Summarization

Background

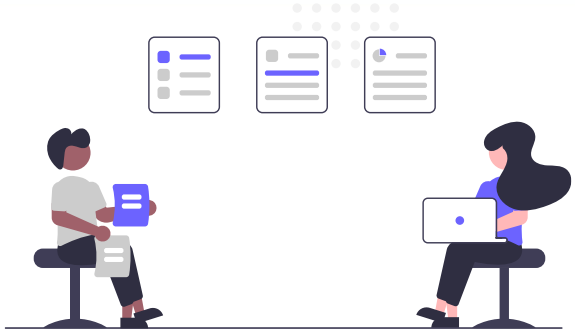
An increase in the number of online meetings made clear that typically meetings only have few key topics and a limited amount relevant information for all participants. Therefore, the extraction of their key topics and their summarization became more obvious. Meetings differ from traditional text as their structure is often dynamic. The interaction between multiple participants (e.g., discussions), their deviant formats, irregular sequences, different semantic styles, and topics promote a complex scenario. Short meetings can easily reach thousands of tokens in just a few minutes of conversation. Thus, techniques that produce high quality meeting summaries, including the most important ideas discussed between its participants, are still necessary.

Goal

- Explore the text summarization task (Extractive/Abstractive) applied to meetings [low resource languages]

Tasks

- Study which models and datasets can be used in this task
- Propose training architecture, training data, or paradigm.
- Evaluate and contribute to state-of-the-art solutions.



Jan Philip Wahle
wahle@gipplab.org

Frederic Kirstein
kirstein@gipplab.org

Terry L. Ruas
ruas@gipplab.org

Semantic Feature Extraction for NLP Tasks

Background

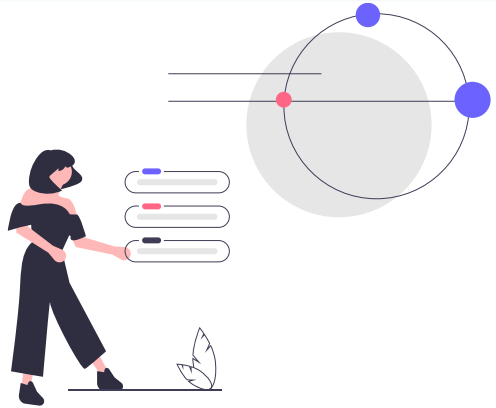
The relationship between words in a sentence often have more semantic content than its actual words individually. Semantic analysis is arguably one of the oldest challenges in Natural Language Processing (NLP) and still present in almost all its downstream applications. However, the extraction of features that describe semantic aspects or the architecture of models/training tasks that capture intrinsic human characteristics is not a trivial task. We are interested in developing methods, training tasks, and architectures that can capture these underlying semantic features and use them in NLP tasks.

Goal

- Develop systems to solve NLP downstream tasks (or defined problems) using semantic features

Tasks

- Review the literature on selected task/problem;
- Extend devised approaches to recent state-of-the-art techniques (propose new ones);
- Evaluate your approach in specific datasets.



Jan Philip Wahle
wahle@gipplab.org

Terry L. Ruas
ruas@gipplab.org

D4: Dynamic DBLP Dataset Discovery

Background

DBLP is the largest open-access repository of scientific articles on computer science and provides metadata associated with publications, authors, and venues. We retrieved more than 6 million publications from DBLP and extracted pertinent metadata (e.g., abstracts, author affiliations, citations) from the publication texts to create the DBLP Discovery Dataset (D3). Now, on D4 we are devising a system (back- and front-end) to explore our dataset and uncover all the trends regarding computer science publications.

Goal

- Develop D4 back and front end – Open issues in our system

Tasks

- Back-end: data loading, pre-processing, database integration, backend client, venue data extraction, (dis)similarity features, etc
- Front-end: responsiveness, back-end integration, filters, etc

Jan Philip Wahle
wahle@gipplab.org

Terry L. Ruas
ruas@gipplab.org