hashformers

Release 1.1.0

Ruan Chaves Rodrigues

Feb 12, 2022

CONTENTS:

CHAPTER

ONE

HASHFORMERS

Open In Colab PyPi license

Hashtag segmentation is the task of automatically adding spaces between the words on a hashtag.

Hashformers is the current **state-of-the-art** for hashtag segmentation. On average, hashformers is **10% more accurate** than the second best hashtag segmentation library (more details on the docs).

Hashformers is also **language-agnostic**: you can use it to segment hashtags not just in English, but also in any language with a GPT-2 model on the Hugging Face Model Hub.

Read the documentation

Segment hashtags on Google Colab

Follow the step-by-step tutorial

1.1 Basic usage

```
from hashformers import TransformerWordSegmenter as WordSegmenter

ws = WordSegmenter(
    segmenter_model_name_or_path="gpt2",
    reranker_model_name_or_path="bert-base-uncased"
)

segmentations = ws.segment([
    "#weneedanationalpark",
    "#icecold"
])

print(segmentations)

# [ 'we need a national park',
# 'ice cold' ]
```

1.2 Installation

Hashformers is compatible with Python 3.7.

```
pip install hashformers
```

It is possible to use **hashformers** without a reranker:

```
from hashformers import TransformerWordSegmenter as WordSegmenter
ws = WordSegmenter(
    segmenter_model_name_or_path="gpt2",
    reranker_model_name_or_path=None
)
```

If you want to use a BERT model as a reranker, you must install mxnet. Here we install **hashformers** with `mxnet-cu110`, which is compatible with Google Colab. If installing in another environment, replace it by the mxnet package compatible with your CUDA version.

```
pip install mxnet-cu110
pip install hashformers
```

1.3 Contributing

Pull requests are welcome! Read our paper for more details on the inner workings of our framework.

If you want to develop the library, you can install **hashformers** directly from this repository (or your fork):

```
git clone https://github.com/ruanchaves/hashformers.git
cd hashformers
pip install -e .
```

1.4 Relevant Papers

- Zero-shot hashtag segmentation for multilingual sentiment analysis
- HashSet – A Dataset For Hashtag Segmentation

1.5 Citation

```
@misc{rodrigues2021zeroshot,
      title={Zero-shot hashtag segmentation for multilingual sentiment analysis},
      author={Ruan Chaves Rodrigues and Marcelo Akira Inuzuka and Juliana Resplande Sant
↪'Anna Gomes and Acquila Santos Rocha and Iacer Calixto and Hugo Alexandre Dantas do␣
↪Nascimento},
      year={2021},
      eprint={2112.03213},
      archivePrefix={arXiv},
      primaryClass={cs.CL}
```

(continues on next page)

(continued from previous page)

```
}
```s

CHAPTER

TWO

# EVALUATION

We provide a detailed evaluation of the accuracy and speed of the `hashformers` framework in comparison with alternative libraries.

Although models based on n-grams such as `ekphrasis` are orders of magnitude faster than `hashformers`, they are remarkably unstable across different domains. Research on word segmentation usually try to bring the best of both worlds together and combine deep learning with statistical methods for reaching the best speed-accuracy trade-off.

## 2.1 Accuracy

In this figure we compare **hashformers** with HashtagMaster ( also known as "MPNR" ) and ekphrasis on five hashtag segmentation datasets.

HashSet-1 is a sample from the distant HashSet dataset. HashSet-2 is the lowercase version of HashSet-1, and HashSet-3 is the manually annotated portion of HashSet. More information on the datasets and their evaluation is available on the HashSet paper.

A script to reproduce the evaluation of ekphrasis is available on scripts/evaluate_ekphrasis.py.

## 2.2 Speed

In this table we evaluate hashformers under different settings on the Dev-BOUN dataset and compare it with ekphrasis. As ekphrasis relies on n-grams, it is a few orders of magnitude faster than hashformers.

All experiments were performed on Google Colab while connected to a Tesla T4 GPU with 15GB of RAM. We highlight `distilgpt2` at `topk = 2`, which provides the best speed-accuracy trade-off.

- **model**: The name of the model. We evaluate ekphrasis under the default settings, and use the reranker only for the SOTA experiment at the bottom row.
- **hashtags/second**: How many hashtags the model can segment per second. All experiments on hashformers had the `batch_size` parameter adjusted to take up close to 100% of GPU RAM. A sidenote: even at 100% of GPU memory usage, we get about 60% of GPU utilization. So you may get better results by using GPUs with more memory than 16GB.
- **accuracy**: Accuracy on the Dev-BOUN dataset. We don't evaluate the accuracy of `gpt2`, but we know from the literature that it is expected to be between `distilgpt2` (at 80%) and `gpt2 + bert` (the SOTA, at 83%).
- **topk**: the `topk` parameter of the Beamsearch algorithm ( passed as the `topk` argument to the `WordSegmenter.segment` method). The `steps` Beamsearch parameter was fixed at a default value of 13 for all experiments with hashformers, as it doesn't have a significant impact on performance as `topk`.

- **layers**: How many Transformer layers were utilized for language modeling: either all layers or just the bottom layer.

CHAPTER

THREE

# API REFERENCE

## 3.1 hashformers

### 3.1.1 hashformers package

**Subpackages**

**hashformers.beamsearch package**

**Submodules**

**hashformers.beamsearch.algorithm module**

**hashformers.beamsearch.bert_lm module**

**hashformers.beamsearch.data_structures module**

**hashformers.beamsearch.gpt2_lm module**

**hashformers.beamsearch.model_lm module**

**hashformers.beamsearch.reranker module**

**Module contents**

**hashformers.ensemble package**

**Submodules**

**hashformers.ensemble.top2_fusion module**

**Module contents**

**hashformers.evaluation package**

### Submodules

### hashformers.evaluation.modeler module

### hashformers.evaluation.utils module

### Module contents

### hashformers.experiments package

### Submodules

### hashformers.experiments.evaluation module

### hashformers.experiments.utils module

### Module contents

### hashformers.segmenter package

### Submodules

### hashformers.segmenter.segmenter module

### Module contents

### Module contents

CHAPTER

FOUR

# INDICES AND TABLES

- genindex
- modindex
- search