Abstract Writing for NLP Papers

Abstract

This is a short guide for abstract writing in NLP papers. The abstract is one of the most important parts of a paper, helping readers quickly understand what the work is about and whether it is worth their time to read. Most authors learn to write good abstracts through experience, but don't synthesize this experience into an explicit pedagogy to help new authors learn quickly. I break down the abstract into several rhetorical components and show how to use them to understand and write abstracts via several examples. Not every abstract needs to adhere to this structure, but this approach can serve as a useful starting point in the abstract-writing process.

So you're writing an NLP paper and you don't know where to start with the abstract or introduction. What's the point of these sections anyway? What's appropriate to include, and how long should it be? How do you get started writing them? In this document, I will try to give a simple starting point for answering these questions.

Purposes of an Abstract

After the title (and maybe some tweets), the abstract is the first thing a reader is likely to see from your paper. It is presented on ArXiv separately from the PDF, appears in BibTeX entries, and may be presented directly in the conference program. This gives it two main (non–mutually-exclusive) functions:

- Briefly summarize the contents and contribution of your paper so that someone who doesn't read it roughly knows what to take away from it and what problems it might be relevant for, and
- Help a reader decide whether they should dig into your full paper.

In addition there are some ancillary functions:

- Serve as input text for keyword searches, for example as performed by people doing meta-analyses, literature review, etc., which are often guided by keyword searches in order to provide the work with a reproducible/documentable methodology and reduce bias
- Serve as highly salient input text for automatic summarization systems like Semantic Scholar's TL;DR functionality, which produces a one-sentence summary of a paper's contribution (more particularly its *what*).
- Probably more I'm not thinking of.

Components of an Abstract

Bearing in mind the high-level purpose of the abstract, we can identify several common components of an abstract by questions they are designed to answer:

- **Background:** What do you need to know in order to understand the rest of the abstract?
- **Content**: What is described/done in the paper?
- **Problem**: What problem is the paper trying to solve?
- **Significance**: Why is this problem important to solve?
- **Challenge**: What makes this problem difficult/non-trivial?
- **Contribution**: What is introduced in this paper in the way of solving the problem?
- **Results:** What observations do you report in the paper?
- **Impact**: What is different (e.g., what is now enabled) because of the work described in the paper?

A straightforward abstract may provide one point for each of these in the exact order given (except background). I'll go through a few examples (mostly from my papers), but it could be a good exercise to annotate one or two on your own, ask what components are present & missing and why. (Not every component is always necessary, and some may deserve more attention for some papers. Also the assignment of sentences to components may be arguable, but that doesn't really matter for the purpose of writing the abstract as long as all the necessary components are addressed.)

Example 1: *Inducing Semantic Roles Without Syntax*

Abstract: Semantic roles are a key component of linguistic predicate-argument structure, but developing ontologies of these roles requires significant expertise and manual effort. Methods exist for automatically inducing semantic roles using syntactic representations, but syntax can also be difficult to define, annotate, and predict. We show it is possible to automatically induce semantic roles from QA-SRL, a scalable and ontology-free semantic annotation scheme that uses question-answer pairs to represent predicate-argument structure. By associating arguments with distributions over QA-SRL questions and clustering them in a mixture model, our method outperforms all previous models as well as a new state-of-the-art baseline over gold syntax. We show that our method works because QA-SRL acts as surrogate syntax, capturing non-overt arguments and syntactic alternations, which are central motivators for the use of semantic role labeling systems.

Example 2: Asking without Telling: Exploring Latent Ontologies in Contextual Representations

Abstract: The success of pretrained contextual encoders, such as ELMo and BERT, has brought a great deal of interest in what these models learn: do they, without explicit supervision, learn to encode meaningful notions of linguistic structure? If so, how is this structure encoded? To investigate this, we introduce latent subclass learning (LSL): a modification to classifier-based probing that induces a latent categorization (or ontology) of the probe's inputs.

Without access to fine-grained gold labels, LSL extracts emergent structure from input representations in an interpretable and quantifiable form. In experiments, we find strong evidence of familiar categories, such as a notion of personhood in ELMo, as well as novel ontological distinctions, such as a preference for fine-grained semantic roles on core arguments. Our results provide unique new evidence of emergent structure in pretrained encoders, including departures from existing annotations which are inaccessible to earlier methods.

Example 3: *Attention is All You Need*

Abstract: The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

A Simple Method for Writing Your Abstract

Just write down a single, independent sentence for each of the components above. Then see how you can string them together to flow naturally.

This method may also apply to writing the introduction — these days, many introductions are written as extended abstracts, covering the same kind of material but in more detail. Other ways of writing introductions may include more expository material, examples, analogies, thought experiments, etc., to orient the reader to a particularly important aspect of the paper's argument. The introduction also often provides (explicitly or implicitly) a road map for the structure of the paper as well.