# The first sentence the second sentence

## a smaller subtitle

Dewi E. Timman
12419273

Bachelor thesis
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*

University of Amsterdam
Faculty of Science
Science Park 900
1098 XH Amsterdam

*Supervisor*
Dr. V. Niculae

Informatics Institute
Faculty of Science
University of Amsterdam
Science Park 900
1098 XH Amsterdam

Semester 1, 2023-2024

# Abstract

**Key words:**

# Contents

# Chapter 1

# Introduction

What if machines can read our mind? If we can give a machine a few key words and let the machine generate a sentence from these key words, our lives would become more productive and efficient. This is what autocomplete systems are trying to achieve. The way in which we choose the key words is also important. Taking just the first or the last few words of a sentence as key words usually does not capture the full meaning of the sentence. For example, if someone want to capture the meaning of 'I live in Amsterdam' in a few key words, the words 'live Amsterdam' would probably be choosen. Thus, the key words come from multiple places in the sentence. Therefore, autocomplete systems need to use more complex models to be more efficient and accurate.

## 1.1 Literature review

### 1.1.1 Autocomplete communication game

The same autocomplete communication game is considered as in Lee et al. (2019). In this game, a human (called user) encodes a sentence into key words. These key words are then decoded by a machine (called system) to retrieve the full, initial sentence. A schematic overview is given in figure 1.1. The communication game is succesfull if the retrieved sentence is the same as the initial sentence.

More formally, a target sentence $x = (x_1, \ldots, x_m)$ is communicated by a user through the key words $z = (z_1, \ldots, z_n)$. Note that $z$ is a subsequence of $x$. The system then tries to retrieve the target sentence by decoding the key words. The target sentence is described by the key words using encoding strategy $q_\alpha(z|x)$ and the system decodes the key words by using decoding strategy $p_\beta(x|z)$.

For a model to be efficient, the number of key words needs to be as low as possible. In addition, for a model to be accurate, the probability of reconstructing

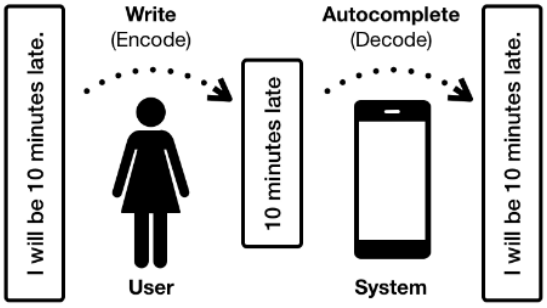


Figure 1.1: schematic overview of the communicatoin game.
Figure from Lee et al. (2019).

$x$ from $z$ needs to be as high as possible. Therefore, we can define a cost and a loss, respectively:

$$\text{cost}(x, \alpha) = \mathbb{E}_{q_\alpha(z|x)}[\text{length}(z)] \tag{1.1}$$

$$\text{loss}(x, \alpha, \beta) = \mathbb{E}_{q_\alpha(z|x)}[-\log p_\beta(x|z)] \tag{1.2}$$

### 1.1.2 Segmentation model

Why does a segmentation model work?

### 1.1.3 Structured latent variables

How does the model work?

## 1.2 Current research

Previous research did not take structure into account (Lee et al., 2019; Bar-Yossef & Kraus, 2011; Svyatkovskiy, Zhao, Fu, & Sundaresan, 2019). Since language is structured, it makes sense to use a structured model as an autocomplete model. In this research, we look at how we can use a latent segmentation model to retrieve key words from a sentence.

# Chapter 2

# Method

# Chapter 3

# Results

# Chapter 4

# Conclusion

# Chapter 5

# Discussion

# References

Bar-Yossef, Z., & Kraus, N. (2011). Context-sensitive query auto-completion. In *Proceedings of the 20th international conference on world wide web* (p. 107-116). ACM.

Lee, M., Hashimoto, T. B., & Liang, P. (2019). Learning autocomplete systems as a communication game.

Svyatkovskiy, A., Zhao, Y., Fu, S., & Sundaresan, N. (2019). Pythia: Ai-assisted code completion system.