Tutoring system
Applied similarity technologies

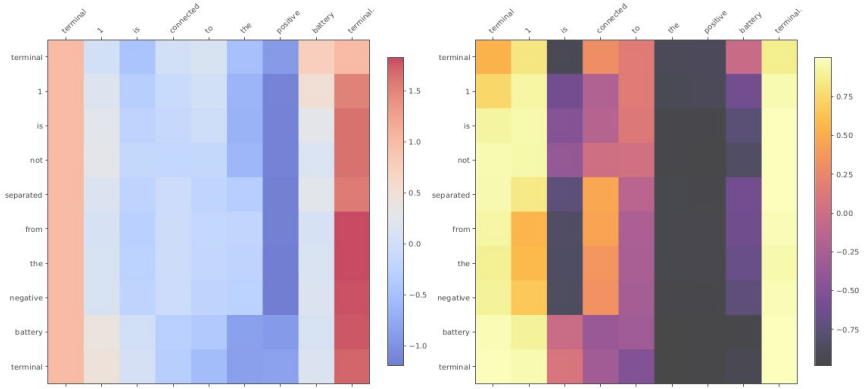
Edgar Andrés
Mohamed Yassin

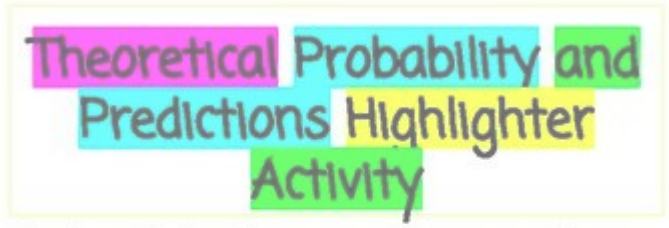
Introduction

- what is a tutoring system?
- why a tutoring system?
- applied similarity to reach tutoring system?

Similarity Technologies

- correlation, alignments and Deep Learning (Scoring)
  - measure similarity in order to estimate the correctness of the student's answer
  - assign grades .
- senses, wordnet , Lesk and combinatory (Feedback)
  - combining the reference and student's answer to generate a correction as feedback
  - highlighting student's mistakes.





Module I

Scoring generation

Automatic Scoring System based on similarity measurements between student and reference answer.

Short Answer Scoring

Short Answer Scoring :

- Treating student’s answers and references in order to detect similarity in between.
- Correctness estimation (training scenario).
- Assign grades. (using external resources Bert similarity)

Scoring based on Similarity

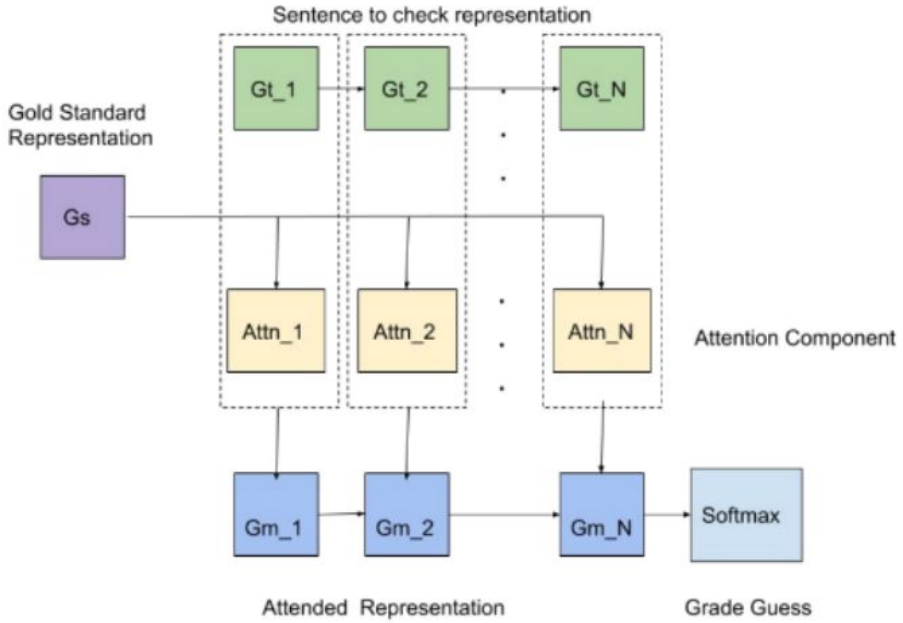
Talking about the neural network

- in: sequences of word embeddings

- RNNs (GRU)

- attention mechanisms

- out: binary correct / incorrect



Updates

- Adjust Correlation measurements
- Reduced Time of process
- Provide Feedback indicator on Incorrect estimation
- Highlighted student answer
- generate Json outputs for interface purposes.

Module II

Feedback generation

The purpose of this process is to extract the main keywords and the most significant words to provide a feedback. the role of the tutor remains necessary.

Feedback generation

the process is made to provide a correction to a **false answer**, spell checker is messing our system, so spelling mistakes can be handled by the system in a wrong way.

The generation of **feedback** is done by combining **student and reference keywords**, this **combination** will lead to a list of **possible indicators** that can form a **correct answer** using students sentence form.

Keywords — Extract most **important words**

Combination — Zip Student & reference **keywords**

Quality — **permutate** the combinations to achieve **best quality** feedback

Highlighting

```
The question was:
Explain why you got a voltage reading of 1.5 for terminal 1 and terminal 6.
The answer was:
Terminal 1 was connected to a positive battery terminal
```

```
The question was:
Explain your reasoning.
The answer was:
bulb b and bulb c are still contained in closed paths with the battery
```

The **highlighting** provide us the information that will **be changed** [**Red**] on the student's answer, this way we can **generate** a **correct feedback** more related to the student answer.

This process is based on the **similarity measures** between **answers**.

Feedback Quality

The **quality of feedback** depends on the answer provided by the student. in our case, we **can not retrieve** related text to provide **more information** [Information Retrieval or knowledge based approaches].

The reference is the key to provide a correct **feedback indicators** and student answer will **be corrected**.

```
was estimated as: correct
was graded as: 7
was really: correct


The feedback was:
good !
The reference was:
Bulb A is still in a closed path with the battery


The question was:
Why not?
The answer was:
switch y is not in the same path of bulb a
```

```
was estimated as: incorrect
was grades as: 3
was really: incorrect


The feedback was:
('bulb if a burns out neither b nor c path are closed paths battery', 4.7091904)
The reference was:
If bulb A burns out, neither B nor C is in a closed path


The question was:
Explain your reasoning.
The answer was:
bulb b and bulb c are still contained in closed paths with the battery
```

Feedback Generation Challenges

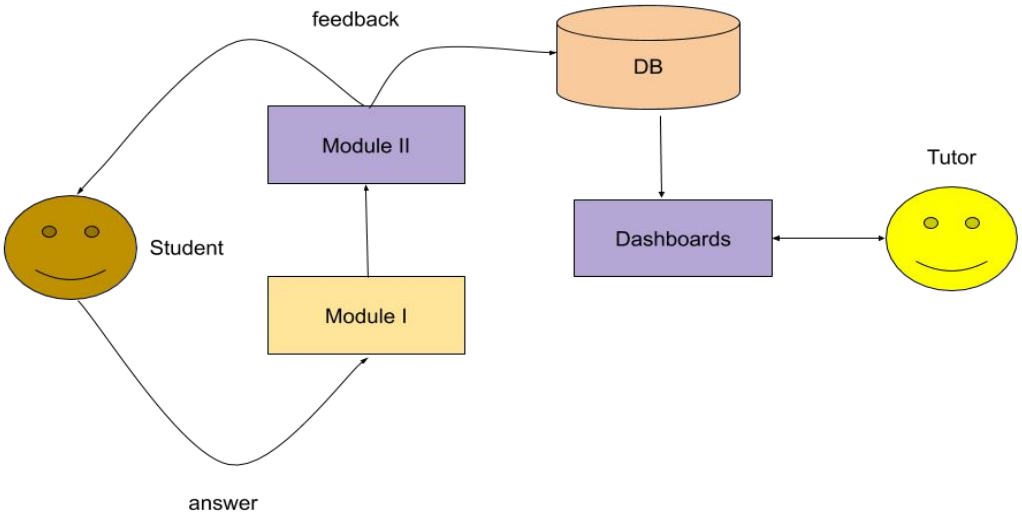
- Spelling mistakes. [**Real experiment Semeval_2013**]
- Extremely low similarity on student answers [**e.g: I don’t know ,tell me the answer**].
- Large reference answers. [**Combinatory Explosion**]
- Incorrect estimation by the module 1. [**Error propagation**]
- Hard to generate a correct sentence as feedback.[**No external resources**]

Prototype

Tutoring system

Building a system that can support a front-end web interface to allow tutors to track student performance.

Design



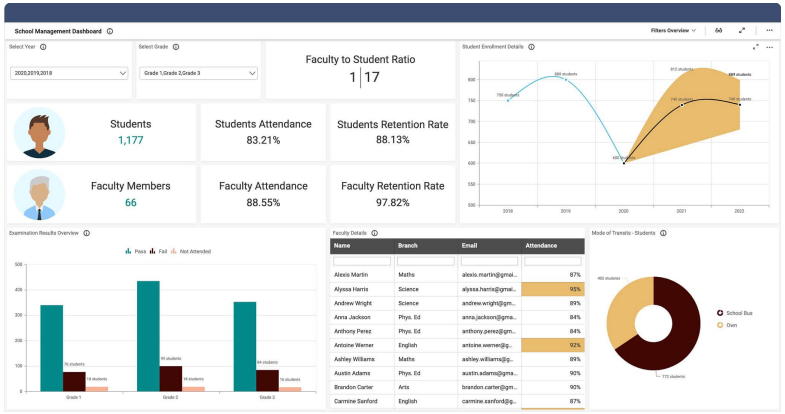
The system depend on two principal **actors**, to **feed** the system with data we need **student's answers** passing by the **1st module** to generate **scores** and then from the **2nd module** we can provide **feedback** to students and also l**oad a database** in order to **implement a dashboard** as web interface for the tutor to **track students performance.**

Dashboards

The main objective of having a dashboard is to track the student performance based on **performance indicators**.

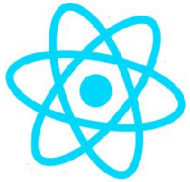
Processing student's data will generate all the data related to their answers [**grades, feedback, Errors ...**].

Data flow should be related to a **web interface** to retrieve more informations about student's task, [**Time, Personal infos,...**]



Dashboards

Dash



React

React, Dash, or Plotly libraries can be used for this purpose, it allow us to create **interactive graphs** to have better representation of our output.

In our case the **output format is Json**, and it's **compatible** to those technologies

Dash also provide a **collection of HTML classes** to generate a **web interface**.

Ethics Agreement

Ideas

- **knowledge acquisition** is the process
- The system purpose is **aid the actors** in the process
- The **tutor** will be always provided the **entire access**
- The **tutor** will always be **needed** in the process

Values

- Equality
- Opportunity
- Justice

Conclusions

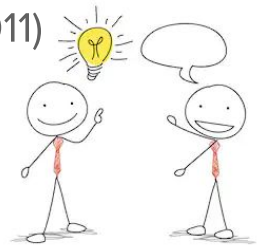
- possibility to **automatize** the process
- the proposed **process** can be **improved**
- the process supports **Architectures** on **top /bottom** (GUI, DB ...)
- this is a **similarity technologies usage**

References

1. Al Emran, M., Shaalan, K.: A survey of intelligent language tutoring systems. In: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI). pp. 393–399. IEEE (2014)

2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

3. Mohler, M., Bunescu, R., Mihalcea, R.: Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 752–762. Association for Computational Linguistics (2011)



References

4. Rocktäschel, T., Grefenstette, E., Hermann, K.M., Kočiskỳ, T., Blunsom, P.: Reasoning about entailment with neural attention. arXiv preprint arXiv:1509.06664 (2015)

5. Wang, S., Jiang, J.: Learning natural language inference with lstm. arXiv preprint arXiv:1512.08849 (2015)

6. Zilio, L., Wilkens, R., Fairon, C.: Using NLP for enhancing second language acquisition. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017. pp. 839–846. INCOMA Ltd., Varna, Bulgaria (Sep 2017). https://doi.org/10.26615/978-954-452-049-6 1 07