*Important points & ToDo’s:*

* Meeting: Monday 5:30 pm
* Updates via Messaging (especially when major problems arise)

Sunggu:

* Preprocessed data without stemming

Seoyoung:

* Checking feasibility of translation via googletrans (1. Step: language detection 2. Step: if detected language <> ENG 🡪 Translate to ENG. For both query & documents)  
  (watching out for exotic languages when detecting the language)

Abi:

* List of languages, potential preprocessing methods for languages <> ENG

Simon & Tom:

* Get rankings (via BM25 & VSM) for 5 queries of good representation, use top ranked documents for each method for pooling

*Notes:*

Tom:

* Lots of probs = 0 (or even < 0)(query == author, stemming) 🡪 Finding reasons, probably preprocessing
  + Stemming rather not

Abi:

* List of languages, preprocessing methods for languages != English, NLTK stop words 23 – but e.g. no Polish
* Consider: Translation of 1 word 🡪 multiple words in target language

Methods for translation:

* Huggin Face: translation + language detection?

Sunggu:

* No stemming at first (query & doc!)

Vsm.py & bm25.py

Agenda:

1. Crosslingual Retrieval / Lecture
   1. **Translation** 🡪 Versions: 1. All to English 2. Translate query into abstract’s language 3. Translate abstract into query’s language
   2. Multilingual knowledge basis: too specific queries 🡪 no
   3. Comparison in dense representation space 🡪 advanced, a.s. no
2. Neural Methods 🡪 advanced, a.s. no
3. Dealing with documents which have no abstract at all 🡪
4. Structure of final .py file:
   1. Raw input dataframe
   2. For each **preprocessing / translation** version one dataframe (common structure!)
   3. **Ranking** methods (currently VSM & different probabilistic approaches [BM25 & Co.]), each producing a ranking-dataframe for each preprocessing version (common structure: **dictionary** **(Tom)** // columns = [‘query-id’, ‘doc-id’, ‘score’ 🡪 1 – 100]. Not far from format required for STELLA – *hand in! / presentation example*)
   4. Current evaluation method: Discounted Cumulative Gain (DCG) 🡪 **Pooling**  
      (Compare different preprocessing- / translation- / ranking- methods)
5. Pooling

Order of translation / preprocessing?

* Tokenization / stop word removal only for English vocabulary?

(Care only about specific single words like ‘Covid-19’, not whether syntax makes sense, or sth. Else..)

🡪 English = target language 🡪 first translate (sentences), then preprocessing

((🡪 English = start language != target language 🡪 first preprocess, then translate (tokens)))

🡪 start language != English != target language 🡪 first translate (sentences) to English, then preprocess in English, then translate (tokens) to target language

**Translation - Versions:**

1. All to English – **priority 1**
2. Query to language of docs (🡪 possible that one query has to be translated into multiple languages)
3. Docs to language of query

(maybe a function which takes 1,2,3 as input and translates accordingly?)

**Preprocessing - Versions:**

* **No stemming!** – **priority 1**
* Stop word removal

**Ranking Model – Versions:**

Queries that are abbreviations: dictionary that maps to full expressions 🡪 scanning docs for both abbreviation & full expression (abbreviations probably also problematic for language detection)

Rankings done by Saturday night, Pooling until Monday (3 or 5 top docs per method; depending on how large the union of them is (should stay <= 10 / 15))

* Check whether googletrans is feasible (first language recognition of query & abstract, if =! English 🡪 translate via API) (Textblob as alternative) (Seoyoung)
* Keep an eye on the list of detected languages (no too exotic languages)
* Dealing with docs without abstract, e.g. weighing doc title (e.g. 10%) & doc abstract (e.g. 90%) – fine tuning

5 queries of good representation 🡪 pooling

Big problems 🡪 messaging

(Quick update on Sunday)

Monday after 5:30 pm (online)

Coaching Session on Wednesday