Strategy

For this specific project, with the limited time and resource, I have chosen unsupervised/ semisupervised learning methods. However, it might be useful when pushing for the final 5% accuracy with supervised learning, given time and resource.

(Coming up with this strategy has been an iterative process, so some of the following could change based on the evaluation results)

ML part:

I have firstly tried FuzzyWuzzy for baseline, a Python package for matching strings based on ranking the number of changes to make to the target string. Immediately, I realised this can not work because the inference time is way too long. So I moved on to develop a basic strategy of tokenisation, encoding and similarity matching as follows:

1. Tokenisation: word piece (spaCy), n-grams (FastText) or bite piece (RoBERTa)
2. Encoding: BM25 ('normalised' TF-IDF),
3. Similarity matching, BM25, NMSLIB
4. Ensemble (See table below for intuition how different methods bring about pro and cons)

	BM25	BM25 + FuzzyWuzzy	FastText	RoBERTa
Pro	Observe unique words very well, useful for identifying key parts of company names	Good at choosing exact match.	Good at handling out of vocabulary words and abbreviations.	• Same as FastText, BPE is good at handling out of vocabulary words and abbreviations • Also good for matching chemist with pharmacist.
Con	• Extremely unique, but irrelevant names can still score high on the list • Exact match can rank lower than other less good matches.	Not giving extra weight to unique key words match. i.e. for Integrum Inc, Integrum Corp. should rank higher than Integrum Hong Kong ltd.	Slow. N-gram encoding takes a long time to complete.	Slow
Improvement strategy	Adding FuzzyWuzzy on top to stop it from mis-rank exact matches		Different data preprocessing: remove non-english companies and subset further by removing legal names that doesn't contain query key words	
Progress				

Testing:

Testing/evaluation can not be exact, because of the lack of labeled data, and the similarity scores vary between different methods. So I put together a list of examples with varying matching difficulties, outlining different challenges (see below) to evaluate performance. This list is compiled in stages to represent the difficult to match examples in each iteration.

Challenges

A quick scan over the query company names and legal names, I have identified the following challenges:

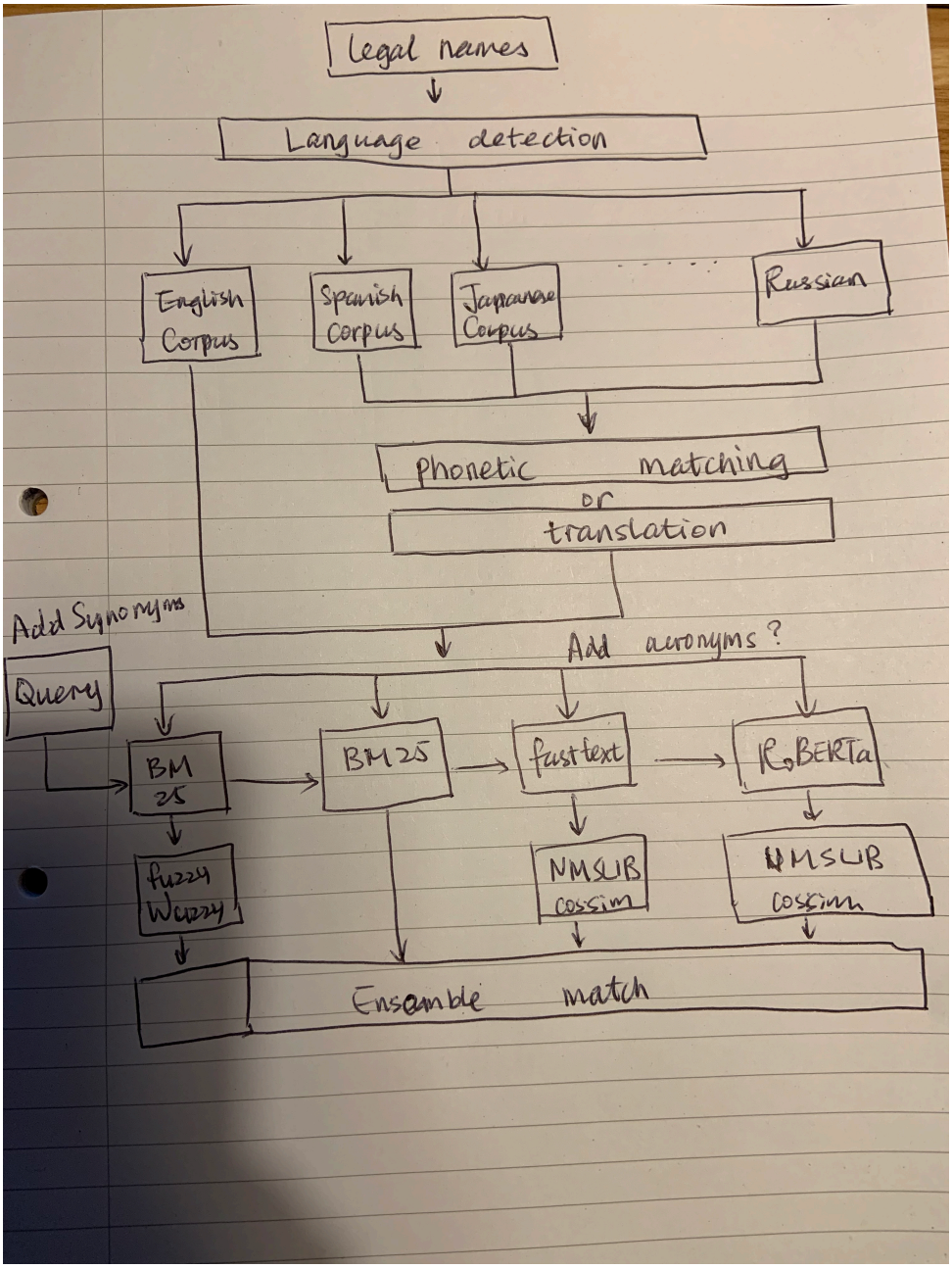
1. Varying prefixes and suffixes: ltd, plc, co., inc.
2. Acronyms: United Overseas Bank, UOB
3. Reordered names
4. Legal names containing branch or subdivision names (requires speaking to the task setter to understand the goal better)
5. Legal names containing alphabetic and non-alphabetic names
6. Missing white spaces

Some possible ways to overcome the challenges

To overcome each challenge, I have come up with these strategies:

5. Stop word removal or word embeddings to take out/reduce the impact of different prefixes and suffixes and word order.
6. Utilise Integrum synonyms (easier to match); transform the legal names to acronyms (harder to match).
7. N-gram encoding to help with missing white spaces and abbreviations
8. Language detection (?) for latinised names, then translation: phonetically (Soundex, Metaphone, Beidor-Morse), or meaning (google translation)

Overall steps



Problems remaining with this approach are:

- The lack of certainty around regional branches and divisions.
- Treatment of match unavailable from the legal dataset, i.e. Burberry. A threshold needs to be set to enable ‘No match found’ instead of ‘THE YOUNG MEN'S CHRISTIAN ASSOCIATION OF METROPOLITAN DENVER’ :)