Human Language Engineering

Practical Work:
BioASQ Challenge Task 6b

Final Presentation

Albert Espín
Lavanya Mandadapu

Contents

01 INTRODUCTION

02 LINGUISTIC DATA ANALYSIS

03 RULE-BASED MODEL

04 MACHINE LEARNING MODEL

05 NEURAL MODELS

06 INCREMENTAL CO-TRAINING WITH QUORA

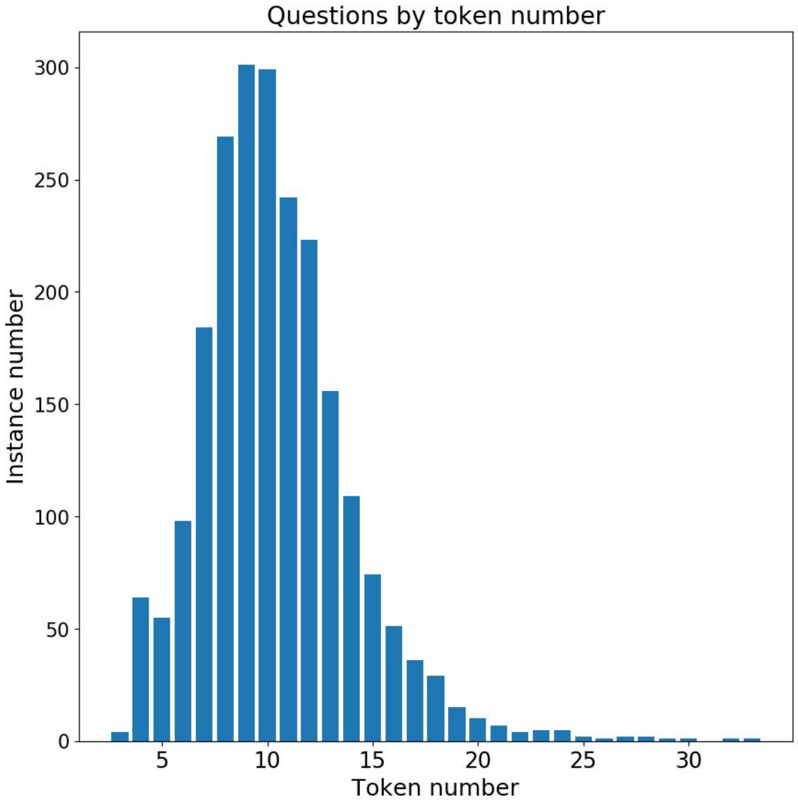
07 CONCLUSIONS AND FUTURE WORK

Introduction

- The practical work is comprised of two tasks:
    - **Task 1**: Development of a multi-class classifier of biomedical questions into one of 4 Question Types:
        - Yes/No
        - Summary
        - List
        - Factoid
    - **Task 2**: Co-training with a broad-domain dataset of Quora questions to try to improve the classification quality.
- **Goals:**
    - Maximize the classification accuracy.
    - Study and compare different models (strengths and weaknesses).

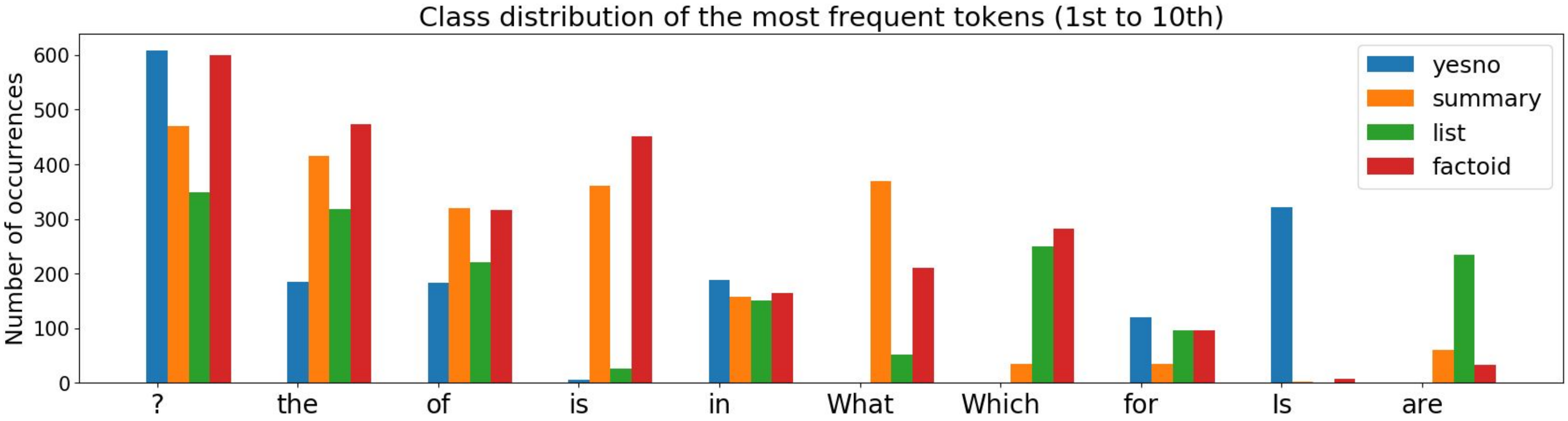
Linguistic Data Analysis

- There are at least 3 tokens per text and a maximum of 33 tokens.
- 97.25% of the texts contains 4-18 tokens. 99.4% questions are single sentences, 12 questions contain 2 sentences, 1 contains 3 sentences.
- The BioASQ corpus is mostly composed of moderately short sentences, making it clear that:
  - Lexical and semantic information will be useful for interpreting the meaning of words to understand the question type.
  - Syntactic information can be useful, since the structure and phrase hierarchy of questions might be indicative of their type.
  - Solving problems like coreference resolution will not be a significant advantage.



Linguistic Data Analysis

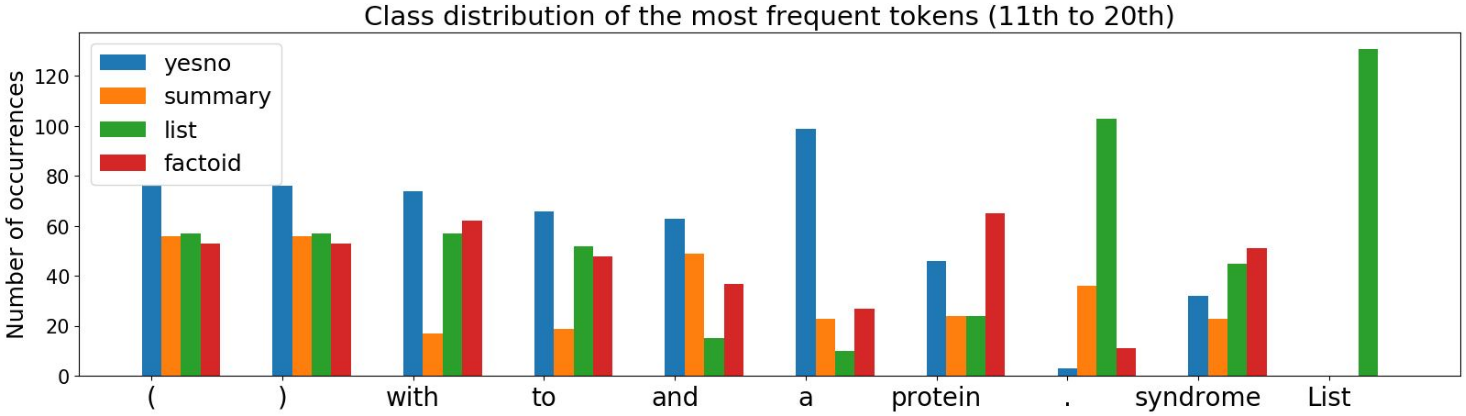
- **Stemming is potentially not useful**, as the class distribution of words with the same stem can be significantly different. For example, “is” is distributed among summary and factoid, while “are” is common in list questions.

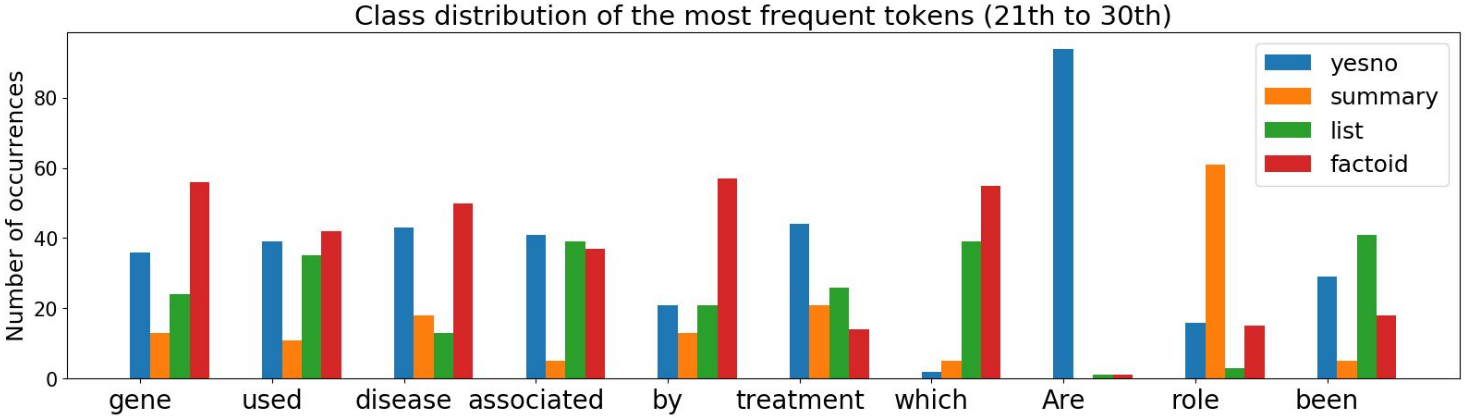


Linguistic Data Analysis

- **Preserving case information can be useful** (at least for the first word), e.g. “Are” clearly appears more in Yes/No questions, while “are” is mostly present in list questions.

- **Words or word-derived features** can help to classify the text.

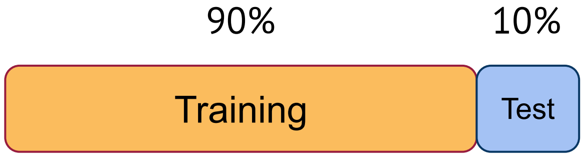




Experimental Setup

- The same data splits are used for all models to perform a fair comparison.
- Initially, the data was shuffled with a fixed seed to remove any hypothetical bias, while preserving reproducibility.
- Stratification was applied to get the same distribution of classes for training and testing.

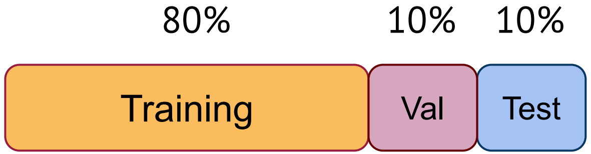
Final classifiers of Task 1:



90% 10%

Training Test

Parameter optimization / Co-training in Task 2

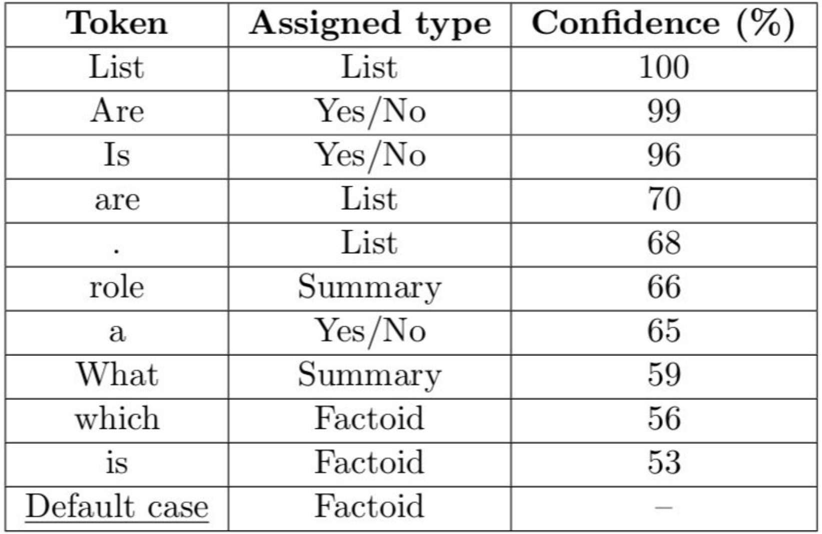


80% 10% 10%

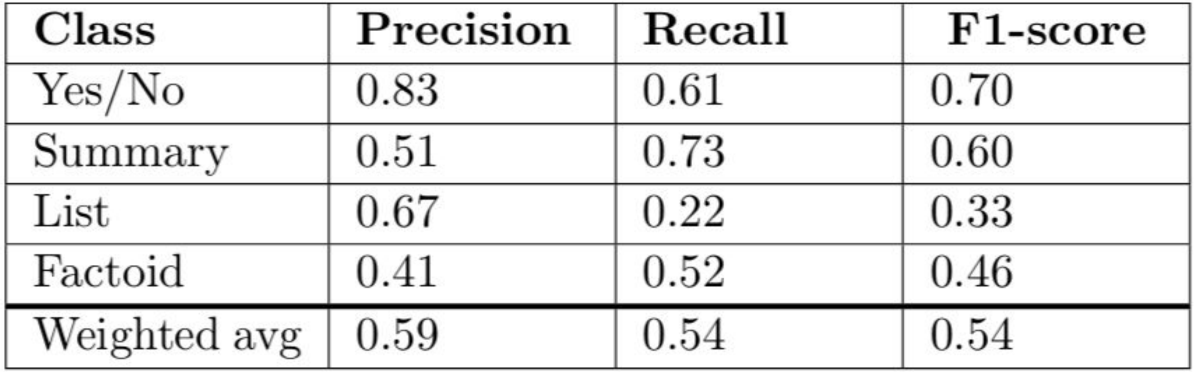
Training Val Test

Rule-Based Model

- The 30 most frequent words (in the training data) are gathered, and those with a dominant class in their distribution are used to design rules.
- Rules are limited to tokens with at least 50% of occurrences for a specific class.
- All question types are represented at least by 1 rule.
- The default case is Factoid (most frequent class), therefore the last two rules are ignored in the final rule set.



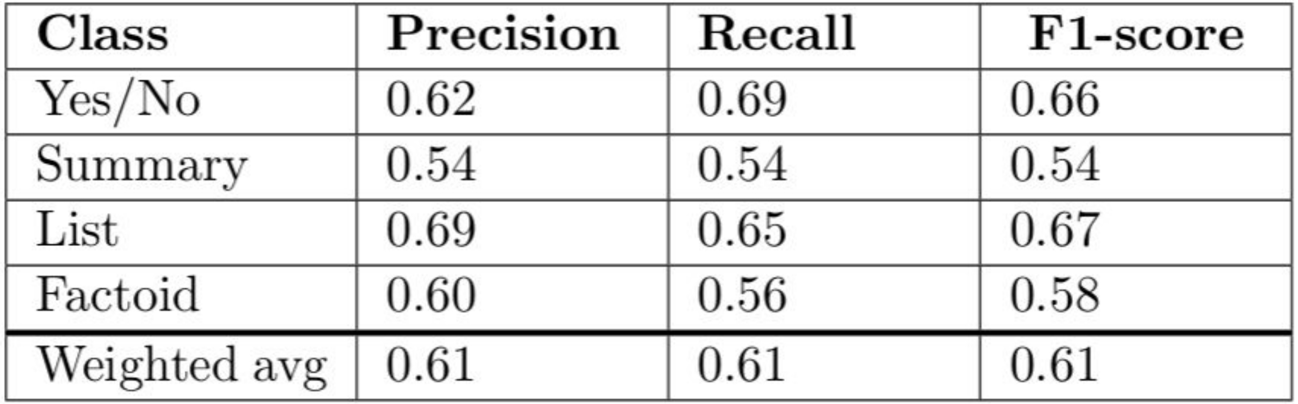
Token	Assigned type	Confidence (%)
List	List	100
Are	Yes/No	99
Is	Yes/No	96
are	List	70
.	List	68
role	Summary	66
a	Yes/No	65
What	Summary	59
which	Factoid	56
is	Factoid	53
Default case	Factoid	–



Class	Precision	Recall	F1-score
Yes/No	0.83	0.61	0.70
Summary	0.51	0.73	0.60
List	0.67	0.22	0.33
Factoid	0.41	0.52	0.46
Weighted avg	0.59	0.54	0.54

Support Vector Machine Model

- Machine Learning model capable of dealing with considerable feature dimensionality, such as word-derived features.
- Binary bag-of-word features discarded due to high orthogonality.
- Words are translated as TF-IDF vectors, to weight the importance of words inversely to their frequency, so that non-stop-words are more valuable.
- SVM model is configured to use a linear kernel after testing Radial Basis and Gaussian kernels on the validation set (which were less-effective).



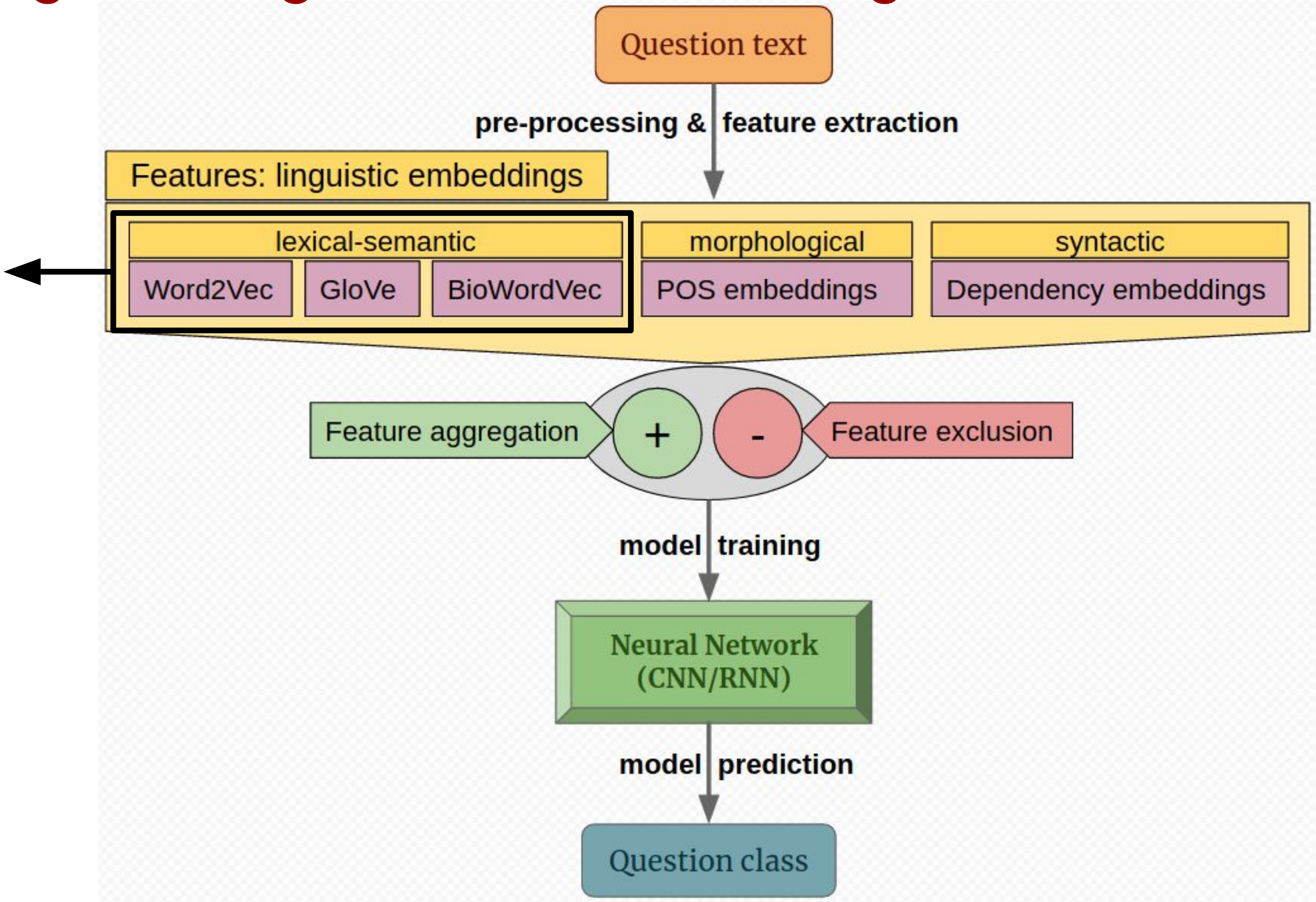
Class	Precision	Recall	F1-score
Yes/No	0.62	0.69	0.66
Summary	0.54	0.54	0.54
List	0.69	0.65	0.67
Factoid	0.60	0.56	0.58
Weighted avg	0.61	0.61	0.61

Neural Network Models - Pre-Processing

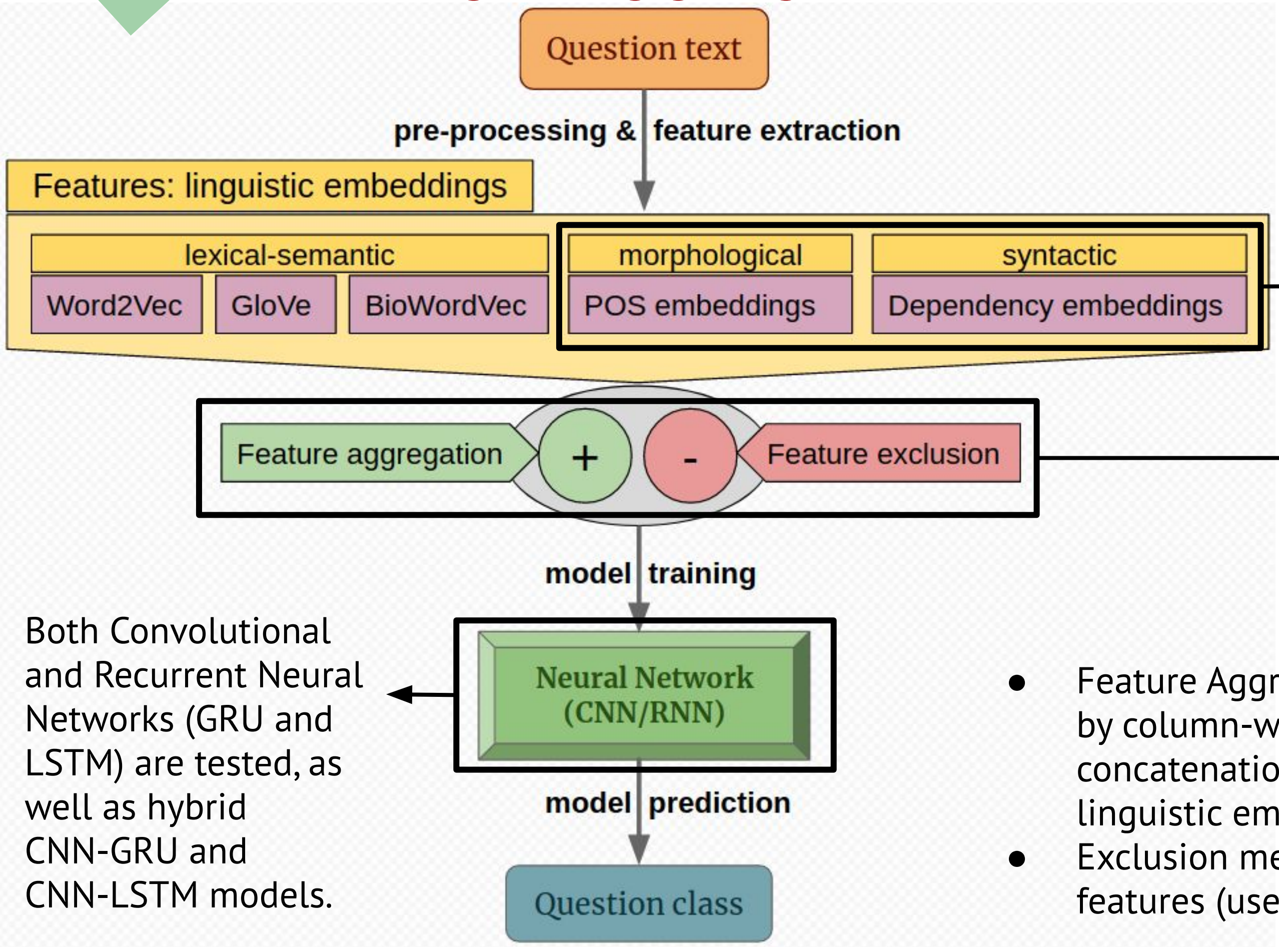
Pre-processing technique	Reason
Lower-casing of all words *Keeping first word capitalized was tested too, but less effective (embedding duality).	Position of the words can be implicitly distinguished by a neural network via activations of convolutional filters.
Preserve dots “ . ” commas ” , ” parenthesis ” (” and ”) ” and question marks ” ? ”.	These add extra information to delimit sentences, clauses and questions, related to syntactic structure.
Consider words with hyphen “ - ” without splitting.	Compound adjectives (e.g. ”well-known”) keep their original notation and are later identified with an independent meaning. Advantageous in the validation set.
Do not remove the stopwords.	Since they connect other words in a syntactically accurate way.

Feature Engineering - Word Embeddings

- **Word2Vec**: These embeddings are trained solely on the vocabulary of our task (4444 tokens).
- **GloVe**: Embeddings trained on broad (Twitter) domain are tested to study its impact on our specific biomedical domain task.
- **BioWordVec**: Domain specific (biomedical) embeddings, fine-tuned with the BioASQ data for even higher adaptation.



Embeddings, Aggregation and Architectures



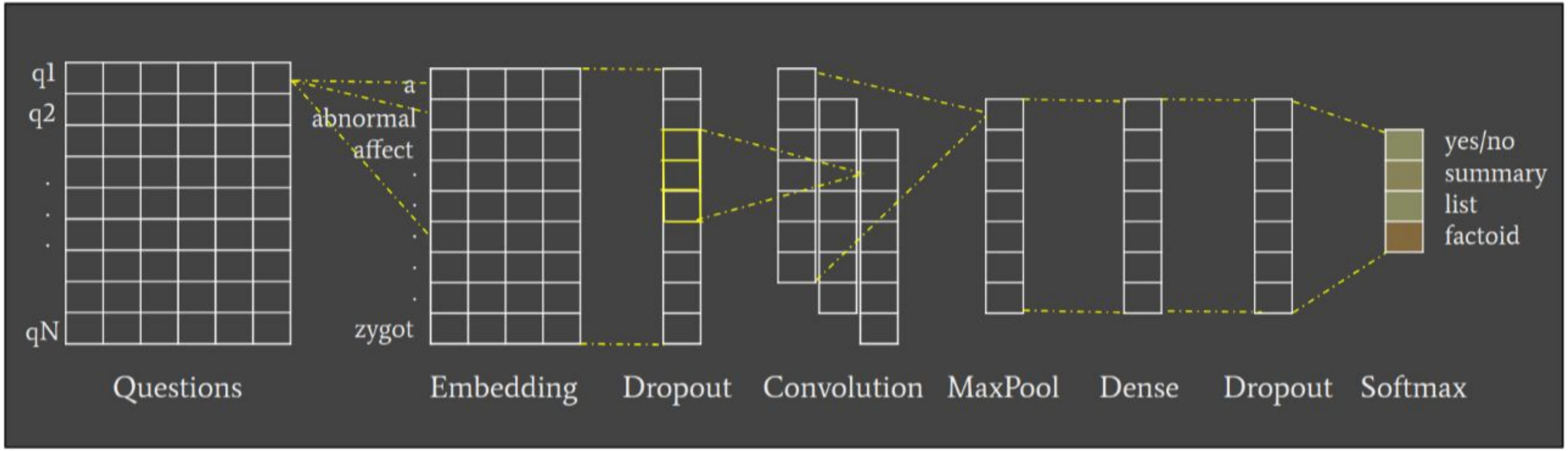
- **POS-tag embeddings**: The morphology of the words can reflect generic patterns to form questions.
- **Syntactic dependency embeddings:** To analyze the syntactic relations of words, we used syntactic triples:
  - *head POS tag*
  - *syntactic relation*
  - *modifier POS tag*

- Feature Aggregation is performed by column-wise vector concatenation of different linguistic embeddings.
- Exclusion means ignoring some features (used in some models).

- Both Convolutional and Recurrent Neural Networks (GRU and LSTM) are tested, as well as hybrid CNN-GRU and CNN-LSTM models.

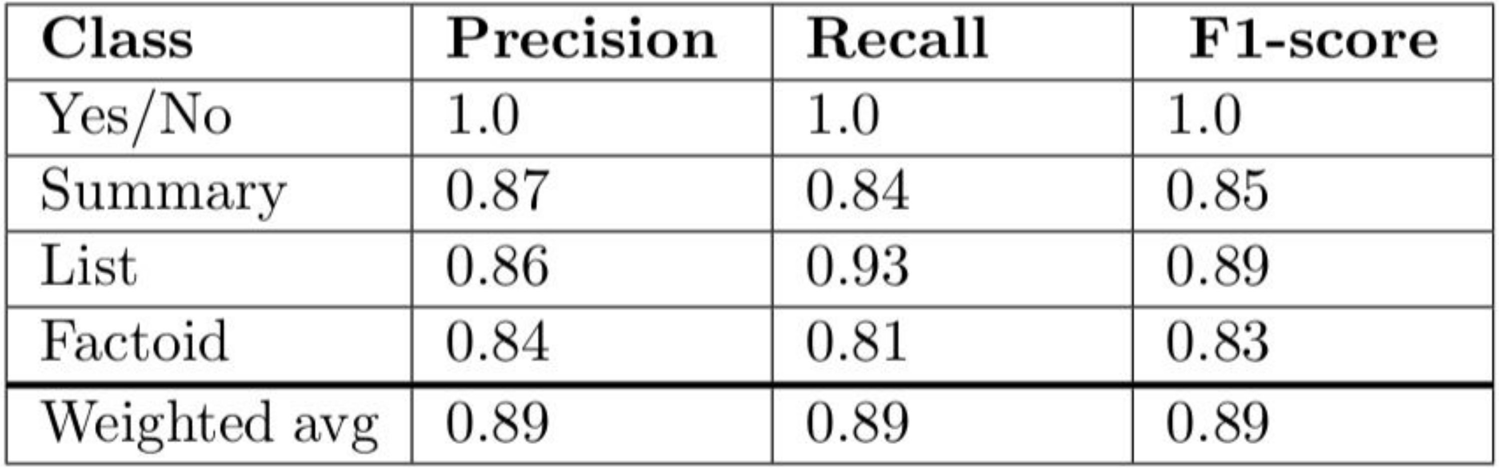
Our Best Neural Network Model - Architecture

- The Convolutional Neural Network model uses 1-D convolution with 300 filters of mask size 5.
- To prevent overspecialization on training data, a dropout ratio of 0.15 is used.
- Global Max-Pooling is added to the architecture to reduce the complexity of the model by keeping the most significant activations followed by the Fully-connected layer with 500 units (for greater predictive power) before the final softmax layer.



Our Best Neural Network Model - Results

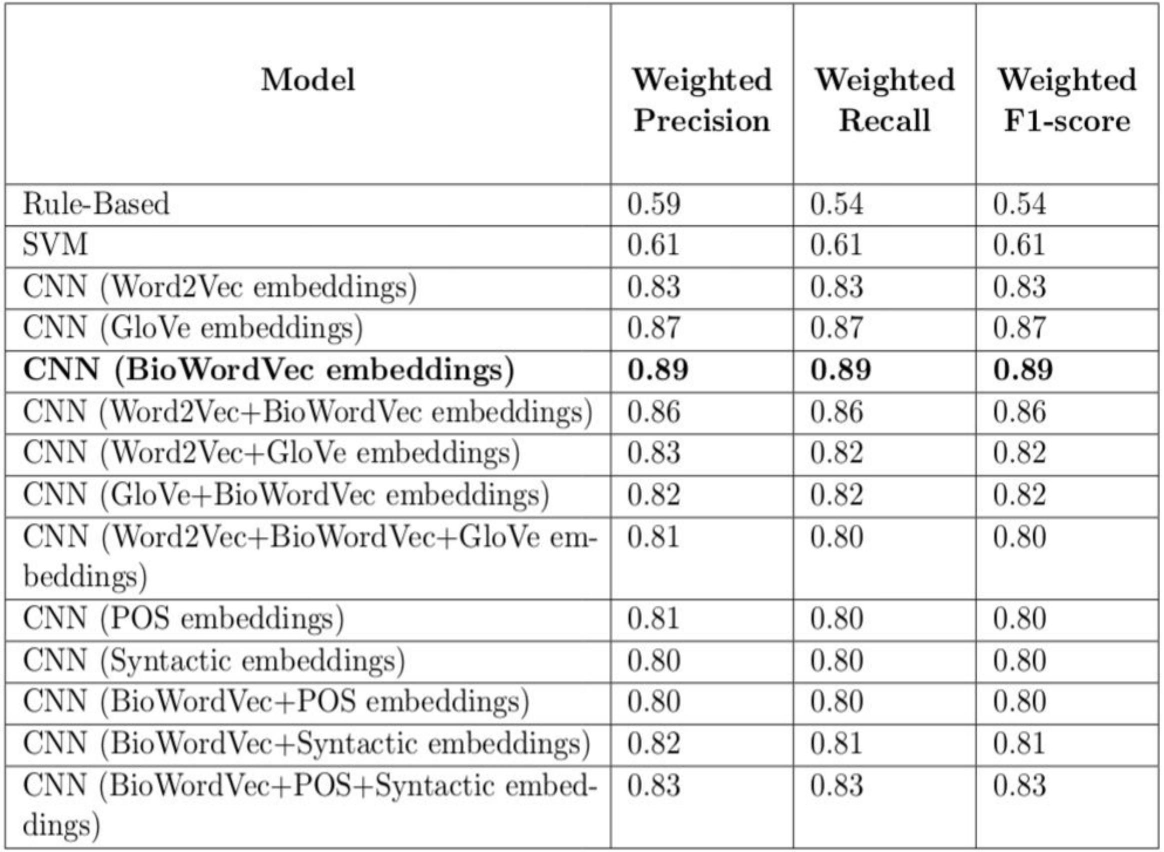
- The CNN model with BioWordVec embeddings has given a high F1 of **0.89**, with **100%** accuracy for the Yes/No question type, due to its simple patterns.
- BioWordVec embeddings exploit biomedical word semantics better.



Class	Precision	Recall	F1-score
Yes/No	1.0	1.0	1.0
Summary	0.87	0.84	0.85
List	0.86	0.93	0.89
Factoid	0.84	0.81	0.83
Weighted avg	0.89	0.89	0.89

Embedding aggregations - Results

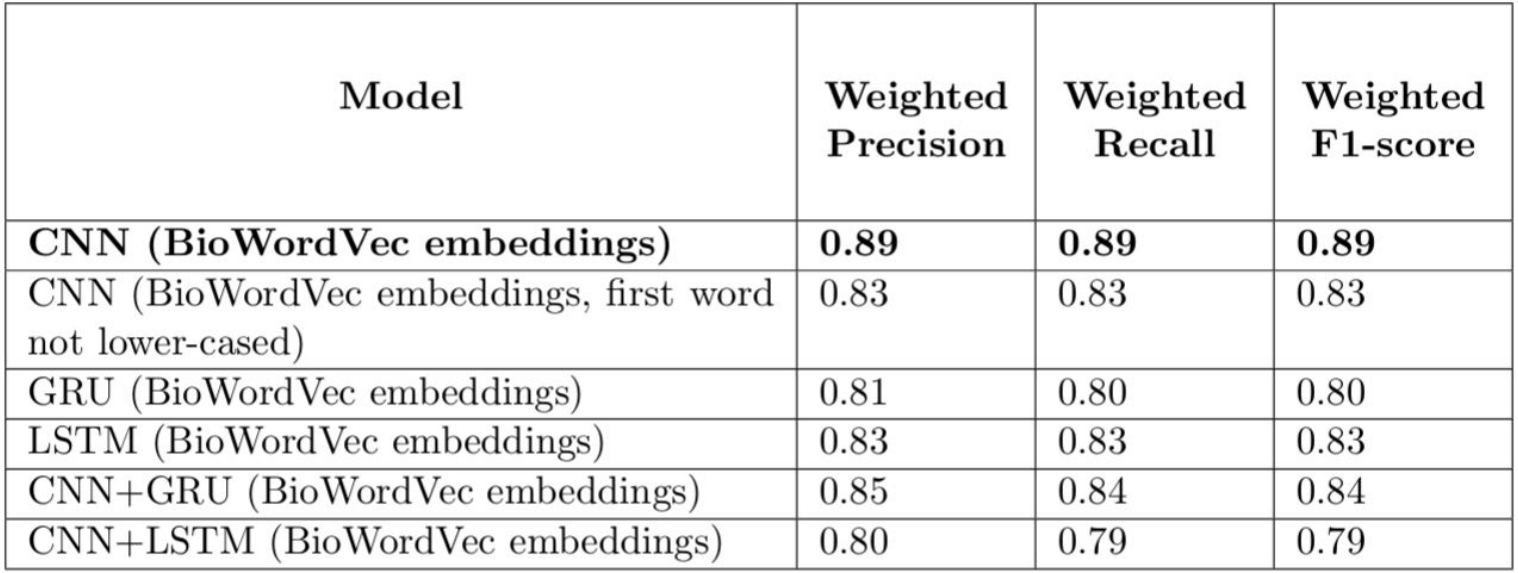
- Aggregating multiple features (embeddings) did not improve the results.
- Different embeddings have different ways of representing the semantics of word vectors.
- A common-space mapping to assimilate all embeddings may have been more effective.



Model	Weighted Precision	Weighted Recall	Weighted F1-score
Rule-Based	0.59	0.54	0.54
SVM	0.61	0.61	0.61
CNN (Word2Vec embeddings)	0.83	0.83	0.83
CNN (GloVe embeddings)	0.87	0.87	0.87
CNN (BioWordVec embeddings)	**0.89**	**0.89**	**0.89**
CNN (Word2Vec+BioWordVec embeddings)	0.86	0.86	0.86
CNN (Word2Vec+GloVe embeddings)	0.83	0.82	0.82
CNN (GloVe+BioWordVec embeddings)	0.82	0.82	0.82
CNN (Word2Vec+BioWordVec+GloVe embeddings)	0.81	0.80	0.80
CNN (POS embeddings)	0.81	0.80	0.80
CNN (Syntactic embeddings)	0.80	0.80	0.80
CNN (BioWordVec+POS embeddings)	0.80	0.80	0.80
CNN (BioWordVec+Syntactic embeddings)	0.82	0.81	0.81
CNN (BioWordVec+POS+Syntactic embeddings)	0.83	0.83	0.83

CNN compared to Recurrent Architectures

- CNN uses convolution to detect local patterns of words that correlate with question types.
- It proves more useful that focusing on data sequentiality and memory, as done by Recurrent Neural Networks, since the BioASQ questions are short and not very complex. This reality is (partially) extended to hybrid CNN-RNN models, and CNN-LSTM is too complex for the small dataset (has overfitting).



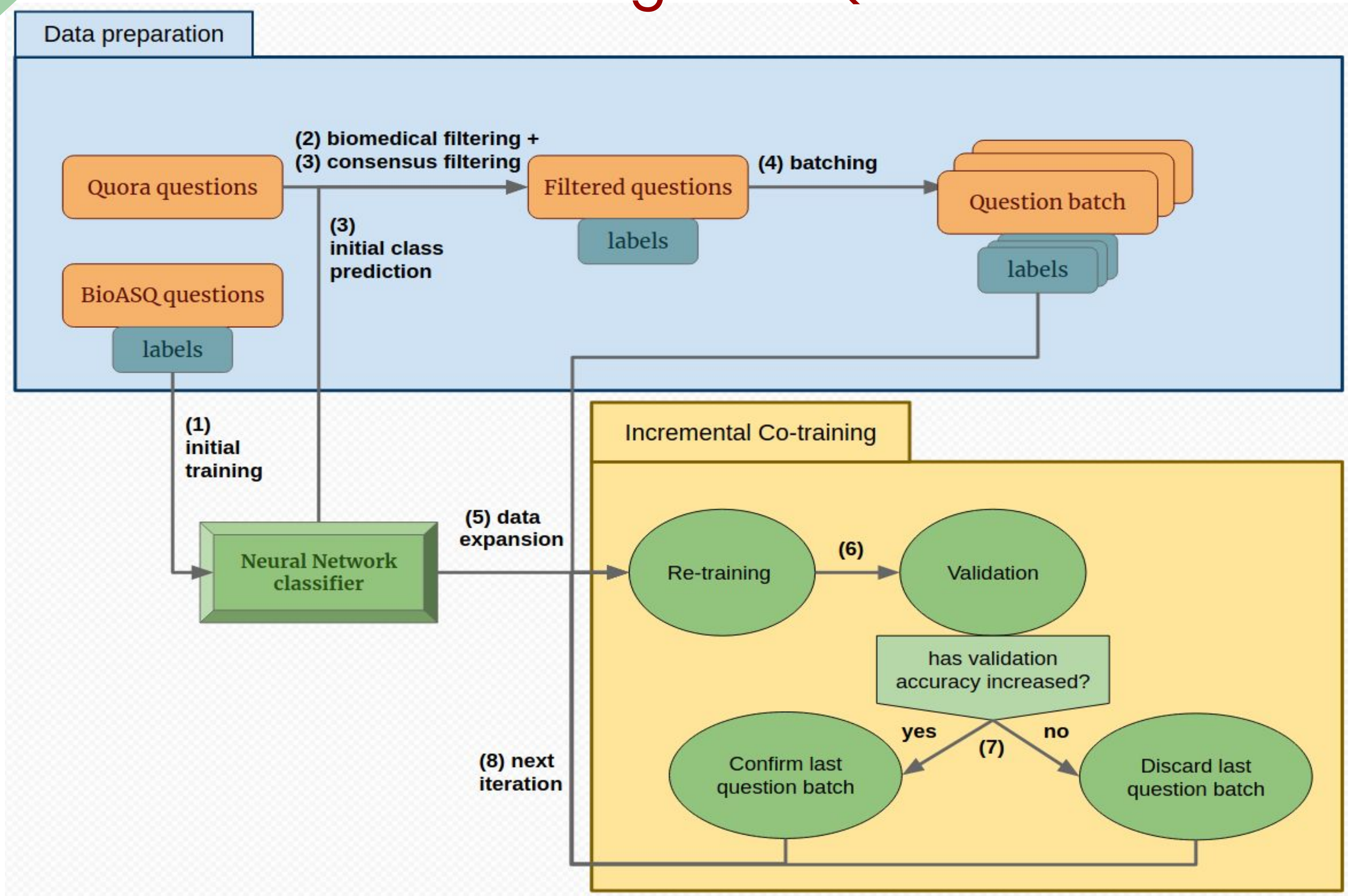
Model	Weighted Precision	Weighted Recall	Weighted F1-score
CNN (BioWordVec embeddings)	**0.89**	**0.89**	**0.89**
CNN (BioWordVec embeddings, first word not lower-cased)	0.83	0.83	0.83
GRU (BioWordVec embeddings)	0.81	0.80	0.80
LSTM (BioWordVec embeddings)	0.83	0.83	0.83
CNN+GRU (BioWordVec embeddings)	0.85	0.84	0.84
CNN+LSTM (BioWordVec embeddings)	0.80	0.79	0.79

Quora dataset - Question filtering

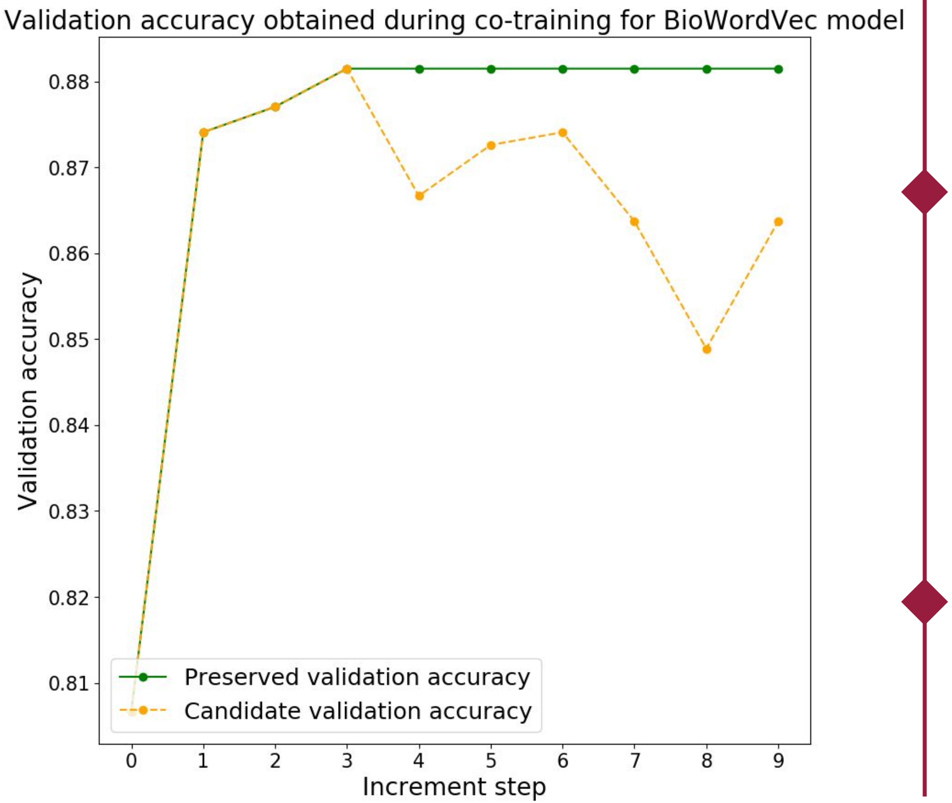
- The quora dataset contains 404K question pairs from different domains.
- For the filtering of biomedical texts, a biomedical dictionary with 100K biomedical terms is considered.
- After the first 2 filters: a total of **8956** question pairs (2% of original data) are accepted.
- In the end: **2822** questions are considered valid to proceed with co-training.

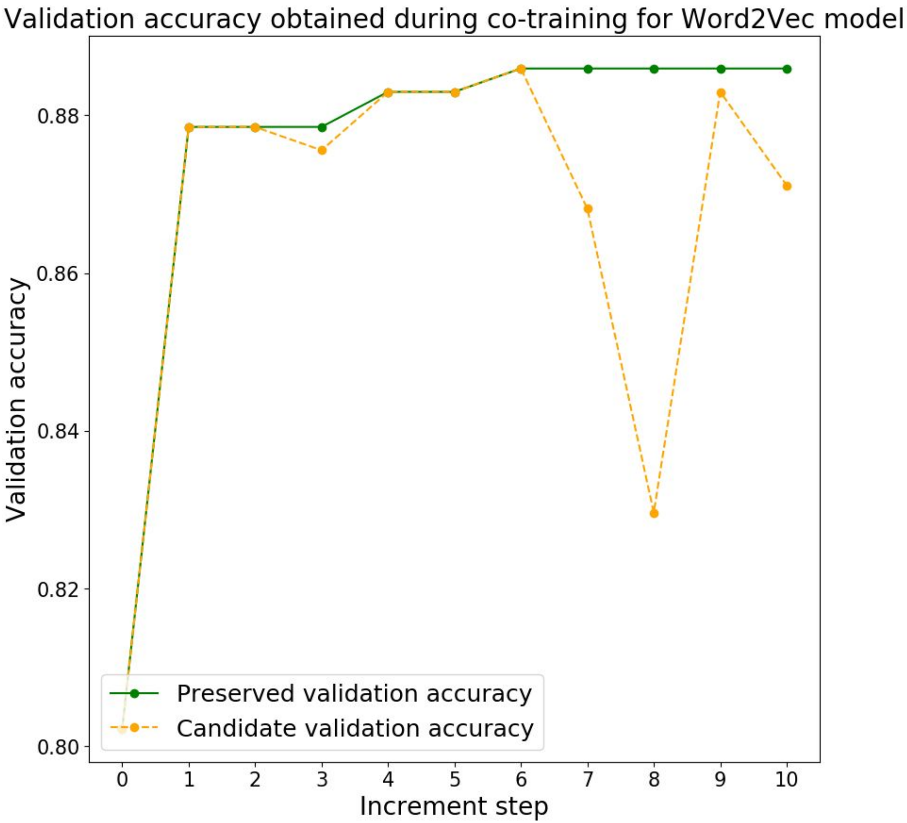
Filter 1	Consider only the questions which contain **38%** of biomedical words in them (ignoring stop-words).
Filter 2	Consider only the questions pairs where both the questions belong to biomedical domain.
Filter 3	Consider the question pairs that has same prediction from the initial CNN model trained on BioASQ (more likely to be correct).
Filter 4	Ignore the questions which are classified as Yes/No since the model already learned to detect them very well.

Incremental Co-training with Quora dataset



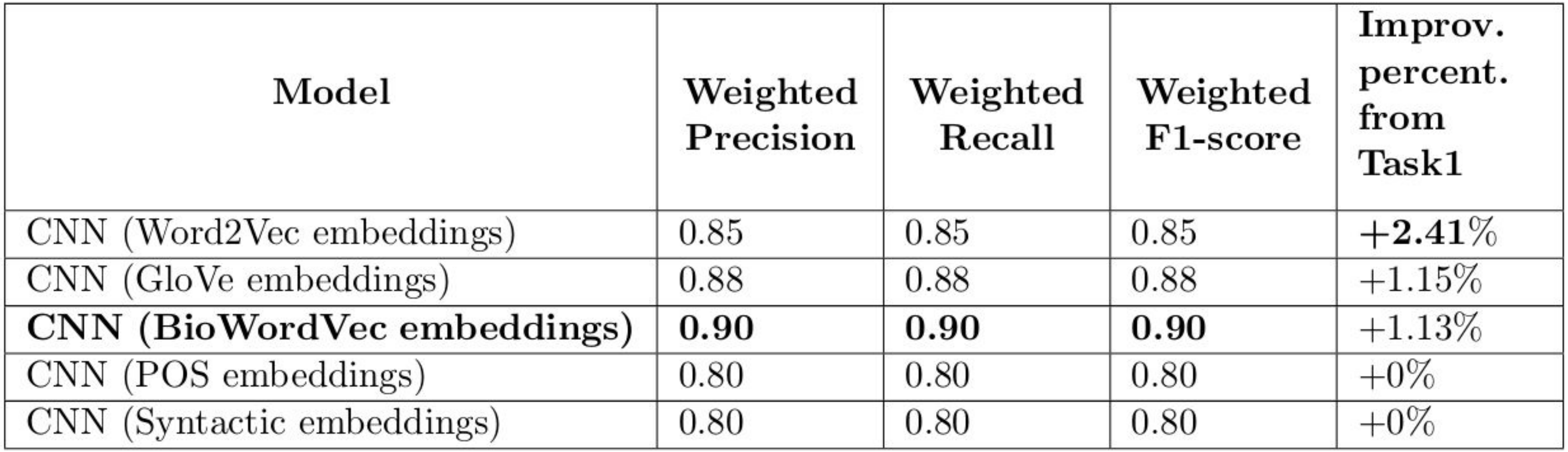
Validation accuracy graphs during co-training





Results of different models after Co-training

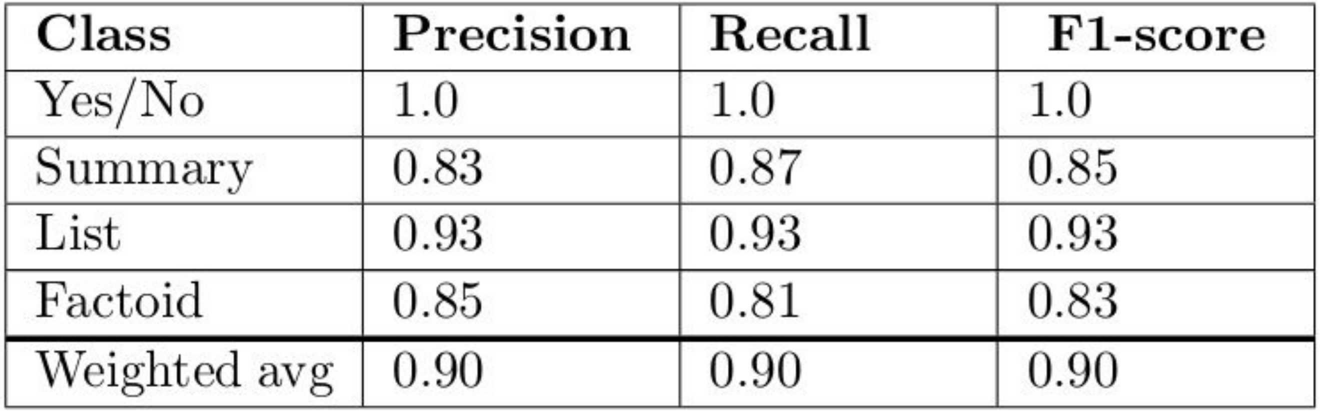
- The model with the highest improvement from Task 1 is the CNN model with Word2Vec features, with 2.41% of increase, because these embeddings were the only ones that had been originally trained in a very small set of texts (the BioASQ questions). All models improve or at least stay equal.
- The filtering strategies discard noisy Quora batches to make the model robust to the inevitable errors in some class assignments for Quora data.



Model	**Weighted Precision**	**Weighted Recall**	**Weighted F1-score**	**Improv. percent. from Task1**
CNN (Word2Vec embeddings)	0.85	0.85	0.85	**+2.41%**
CNN (GloVe embeddings)	0.88	0.88	0.88	+1.15%
CNN (BioWordVec embeddings)	**0.90**	**0.90**	**0.90**	+1.13%
CNN (POS embeddings)	0.80	0.80	0.80	+0%
CNN (Syntactic embeddings)	0.80	0.80	0.80	+0%

Our Best model with Co-Training - Results

- The best CNN model of Task 1 (with BioWordVec embeddings) is also the best in Task 2.
- The F1-score has slightly improved up to **0.90**, mainly due to the introduction of many questions of the list type, which was the least common in BioASQ data.
- The other types have the same F1: having more data is good, but Task 1 model's sometimes confuses summary with factoid for Quora questions, propagating error.



Class	Precision	Recall	F1-score
Yes/No	1.0	1.0	1.0
Summary	0.83	0.87	0.85
List	0.93	0.93	0.93
Factoid	0.85	0.81	0.83
Weighted avg	0.90	0.90	0.90

Conclusions and Future Work

- The usage of Convolutional Neural Network models with word embeddings has proved very effective to learn characteristic local word patterns for classification.
- BioWordVec embeddings as features, fine-tuned with the BioASQ data for greater adaptation, lead to reach a F1-score of **0.89** in Task 1.
- In Task 2, incremental learning guided with validation accuracy provides a final F1-score of **0.90**, slightly better.
- Future Work to refine the classifier for the BioASQ challenge may include the addition of more data to learn the different question types more precisely.
- However, the quality of the annotation is key, otherwise leads to error propagation. Manually annotated data would be ideal.
- Data suitability (i.e. domain matching) is also important. Instead of the filtering process followed in this work, another option is to build a binary classifier to detect if a text is from biomedical domain or not.

Do you have any questions?

Thank you