A Neural Candidate-Selector Architecture for Automatic Structured Clinical Text Annotation

G. Singh[1] I. Marshall[2] J. Thomas[1]
J. Shawe-Taylor[1] B. C. Wallace[3]

[1]Department of Computer Science
University College London, London

[2]Department of Computer Science
Kings College London, London

[3]College of Computer Science
Northeastern University, Boston

CIKM, 2017

Problem: Text Annotation

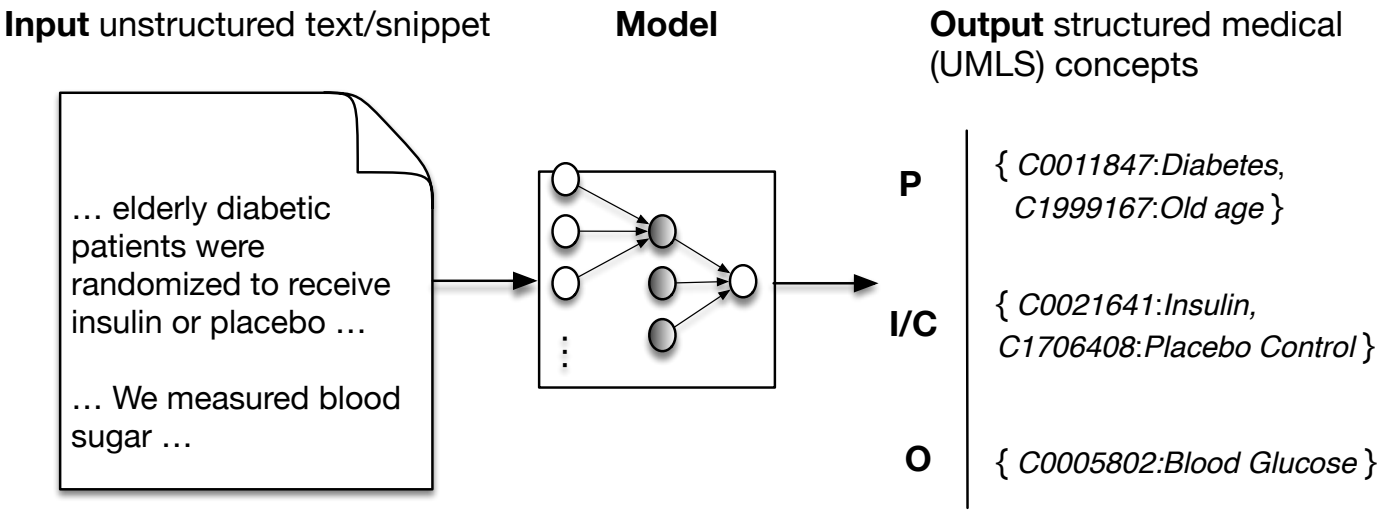


Figure: Illustration of the annotation task.

Problem: Clinical Trial Annotation

- Each clinical trial needs to be associated to multiple labels
- Labels are drawn from a controlled medical vocabulary
- Labels are associated to following disjoint categories:
  - Population
  - Intervention/Control
  - Outcome
- Vast output space for prediction
- Limited labelled training data

Motivation: Clinical Trial Annotation

- Correct labels are crucial during search for relevant literature
- Risk-of-bias is reduced with precise annotation
- Requires extensive human-effort to annotate the trials
- Takes time to annotate labels manually

Method: Candidate-Selector Architecture

- **Candidate Generation**: We generate candidate labels with high recall, to be passed on to the selector
- **Candidate Selection**: We prune the candidate set to obtain labels with high precision

Candidate Selection: Deep Model

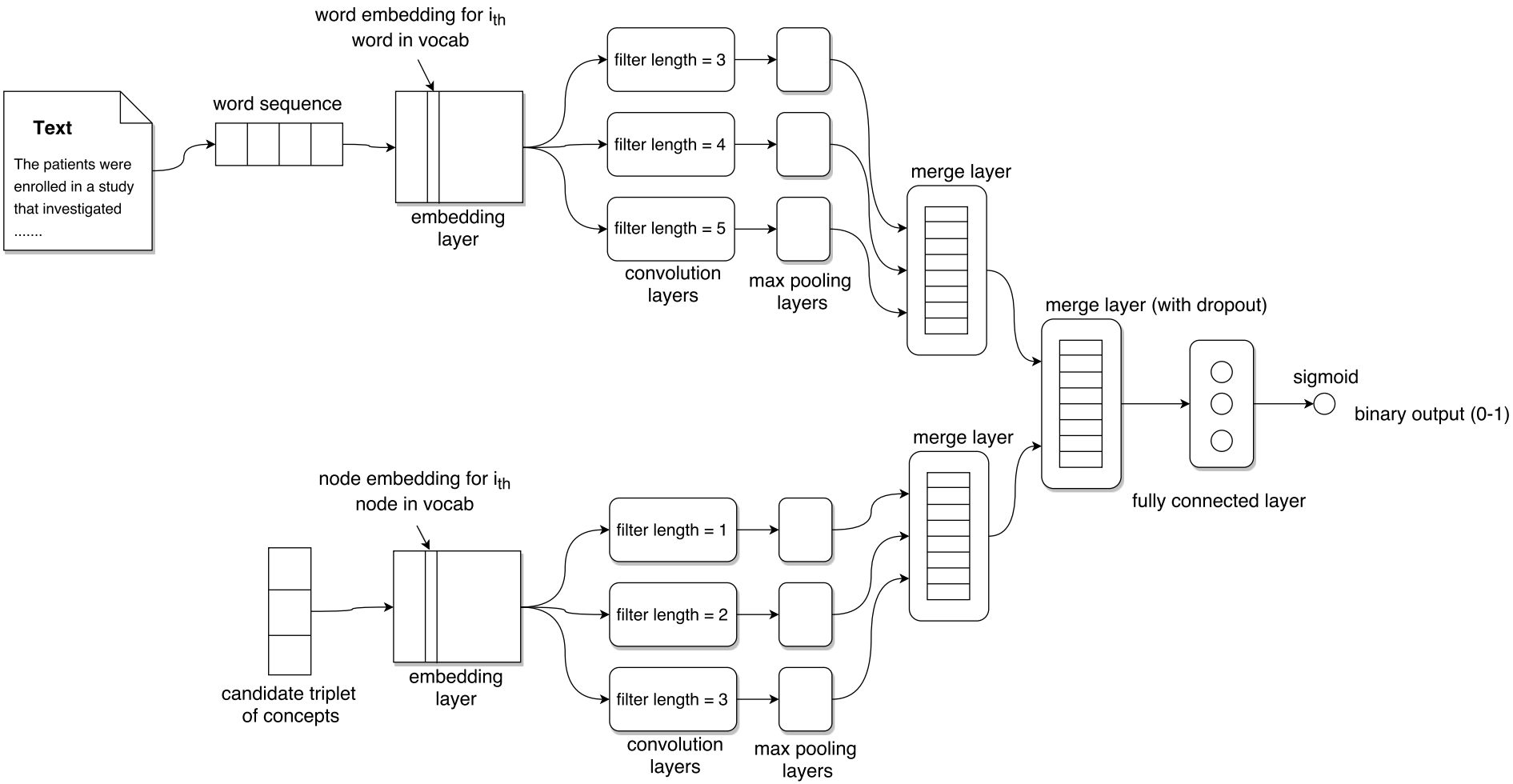


Figure: Candidate Selector Model

Candidate Selector: Model Training

- The model inputs during training are:
  - Free text (or abstract) that needs to be annotated
  - Candidate label triplets obtained for the given text
- Model output is a binary decision $\{0,1\}$
- Two types of triplets are constructed:
  - Positive triplets
  - Negative triplets

Candidate Selector: Model Training

- Positive triplets:
  - Formed by choosing one label each from P, I/C and O
  - Constructed from the ground truth annotations for P, I/C and O
  - All possible triplets are constructed using the given annotations.
- Negative triplets:
  - Non-existent triplets are constructed as negative samples
  - Help the model distinguish between relevant and irrelevant triplets

Candidate Selector: Model Training

- Two types of triplets used:
  - Complete triplets of the form $(c_P, c_{I/C}, c_O)$
  - Incomplete triplets of the form $\{(c_P, \_, \_), (c_P, c_{I/C}, \_)\}$
- Complete triplets used to learn joint distribution of labels
- Incomplete triplets useful for learning of marginals
- Ratio of # of positive triplets to # of negative triplets is 1:5

Candidate Generation

- At test time:
  - We generate candidates for each of the categories i.e. P, I/C and O
  - Using following two methods:
    - Metamap: It is a software deigned over the years to generate a list of concepts associated to a clinical text.
    - Multitask learning: We train a traditional multi-task deep learning model that can predict possible candidates during test time for a given text.

Metamap Service

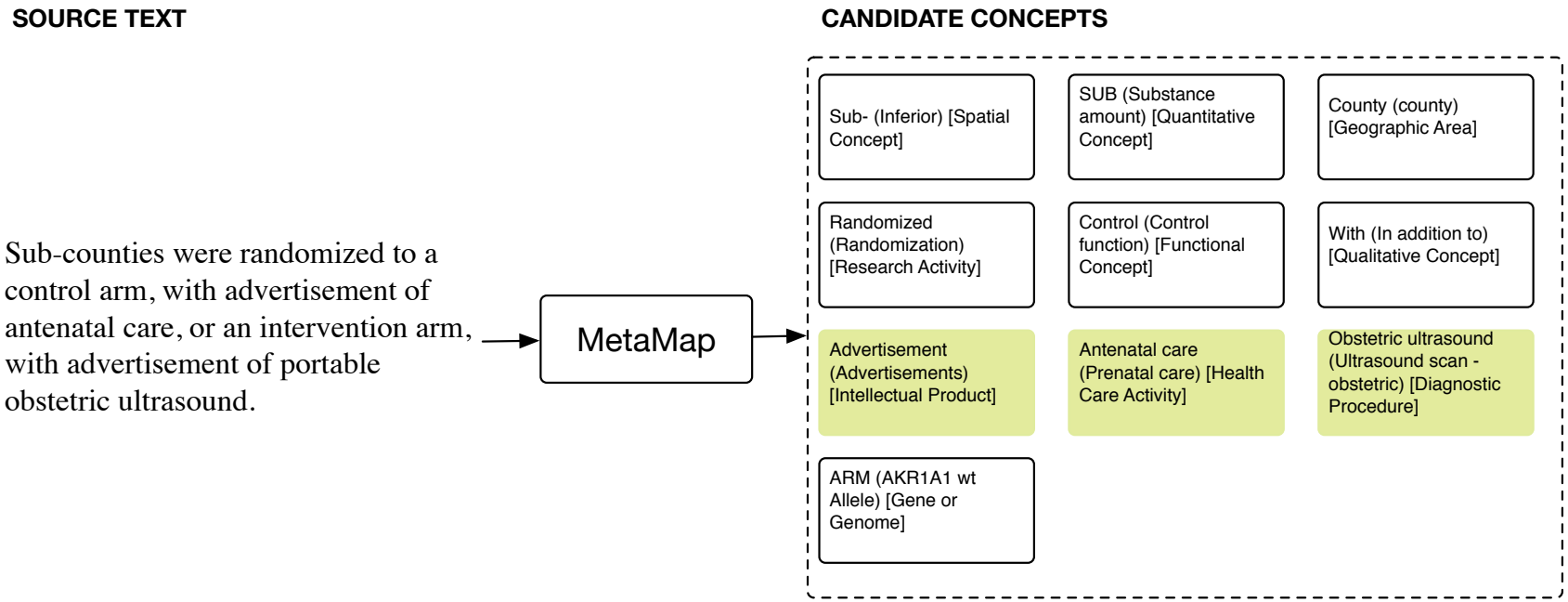


Figure: Metamap Service

- Returns a list of concepts based on the text
- Generates concepts in an unsupervised way
- Returns relevant concepts not seen during training

Multitask Model

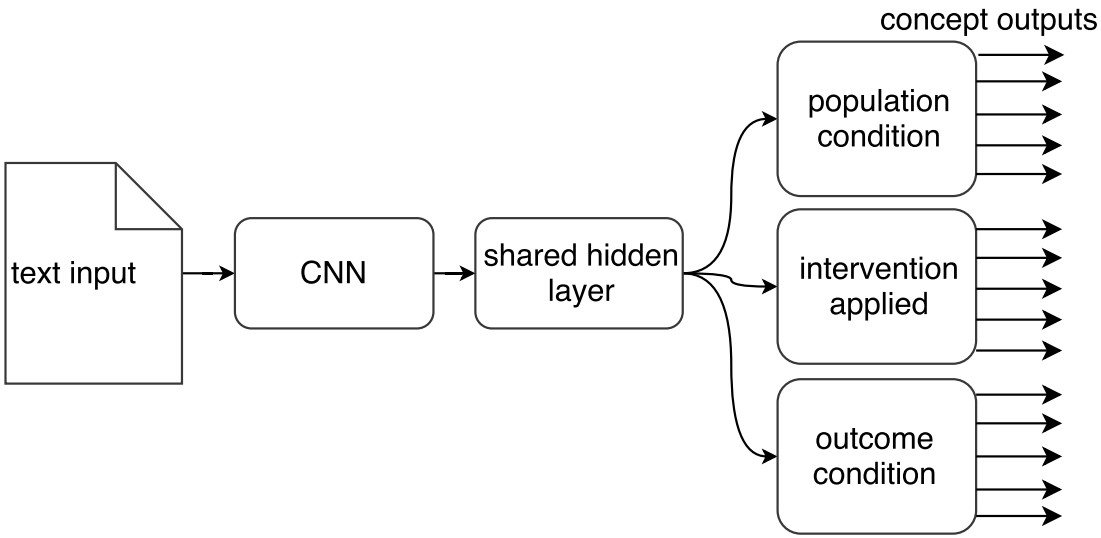


Figure: Multitask Model

- Used to generate candidate terms from those seen in the training set
- Provides a precise list of candidate terms that are relevant to the text

Dataset: Statistics

samples (clinical trials)	4306
distinct population concepts	875
distinct intervention concepts	1115
distinct outcome concepts	1731
population concepts	9387
intervention concepts	5458
outcome concepts	13800

Table: Dataset statistics.

Dataset: Characteristics

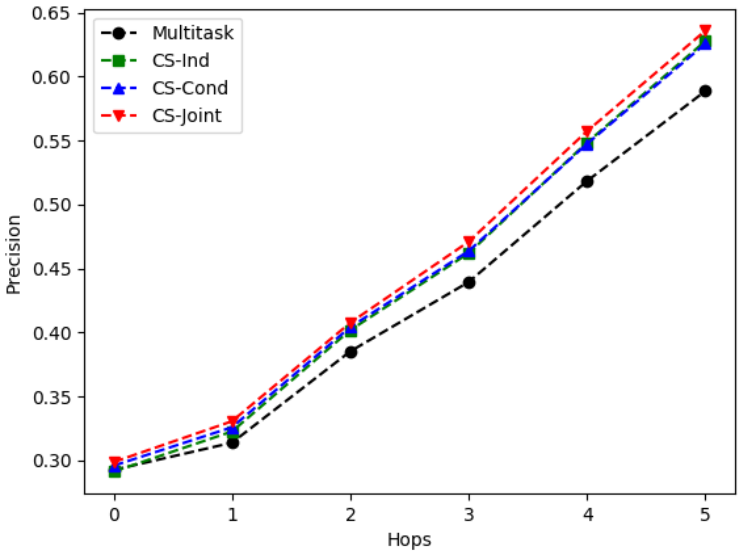
- Dataset consists of text describing clinical trials
- Concepts drawn from a restricted Unified Medical Language System (UML) vocabulary corresponding to PICO elements
- The PICO element corresponds to Population, Intervention/Control and Outcome
  - P: **P**opulation concerns the characteristics shared by trial participants (e.g., diabetic males)
  - I/C: **I**nterventions are the active treatments being studied (e.g., aspirin)/**C**omparators are baseline or alternative treatments to which these are compared (e.g., placebo).
  - O: **O**utcomes are the variables measured to assess the efficacy of different treatments (e.g., headache severity)

Experimental Results

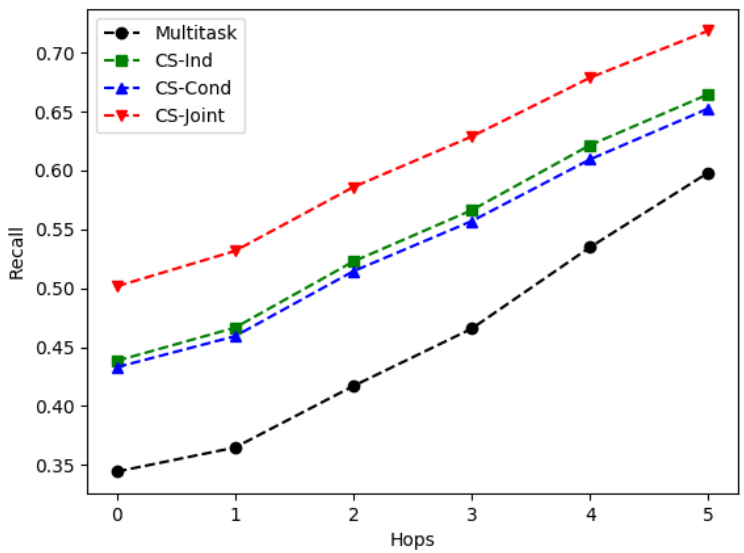
Category	Model	Precision	Recall	F1-score	Pr-2hops	Re-2hops	F1-2hops
Population	MetaMap	0.134	0.280	0.181	0.262	0.489	0.341
	Multitask	0.358	0.383	0.370	0.501	0.502	0.501
	CS-Ind	**0.385**	0.529	0.446	**0.557**	0.636	**0.594**
	CS-Cond	0.384	0.535	**0.447**	0.553	0.640	0.593
	CS-Joint	0.318	**0.594**	0.415	0.485	**0.709**	0.576
Interventions/Comparator	MetaMap	0.108	0.288	0.157	0.163	0.387	0.230
	Multitask	0.248	0.245	0.246	0.264	0.262	0.263
	CS-Ind	0.226	0.272	0.247	0.274	0.322	0.296
	CS-Cond	0.225	0.282	0.250	0.275	0.331	0.300
	CS-Joint	**0.265**	**0.421**	**0.326**	**0.314**	**0.473**	**0.378**
Outcomes	MetaMap	0.209	0.391	0.273	0.314	0.518	0.391
	Multitask	0.198	0.211	0.204	0.283	0.290	0.286
	CS-Ind	0.272	0.497	0.352	0.380	0.593	0.464
	CS-Cond	0.268	0.497	0.348	0.378	0.591	0.461
	CS-Joint	**0.279**	**0.503**	**0.359**	**0.38**	**0.595**	**0.468**

Table: Precisions, recalls and f1 measures realized by different models on the respective PICO elements.

Experimental Results



(a) Precision



(b) Recall

Figure: Average (over PICO elements) r-precisions (a) and recalls (b) for each method as a function of r, counts a predicted concept as matching the truth concept when it is $\leq r$ hops away.

Experimental Results

Category	Model	Precision	Recall	F1-score	Pr-2hops	Re-2hops	F1-2hops
Population	CS-Joint random	**0.268**	**0.251**	**0.259**	0.386	0.382	0.384
	CS-Joint pre-trained	0.264	0.250	0.257	**0.392**	**0.392**	**0.392**
Interventions/Comparator	CS-Joint random	0.219	0.248	0.233	0.272	**0.294**	**0.283**
	CS-Joint pre-trained	**0.233**	**0.257**	**0.244**	**0.273**	0.293	0.282
Outcomes	CS-Joint random	0.315	0.302	0.308	0.412	0.404	0.408
	CS-Joint pre-trained	**0.341**	**0.356**	**0.348**	**0.440**	**0.449**	**0.445**

Table: The performance of the CS-Joint model when using randomly initialized versus pre-trained embeddings.

Experimental Results

Category	Model	Precision	Recall	F1-score	Pr-2hops	Re-2hops	F1-2hops
Population	CS-Joint Complete	0.197	0.145	0.167	0.267	0.216	0.239
	CS-Joint +Marginals	**0.264**	**0.250**	**0.257**	**0.392**	**0.392**	**0.392**
Interventions/Comparator	CS-Joint Complete	0.156	0.149	0.153	0.180	0.168	0.174
	CS-Joint +Marginals	**0.233**	**0.257**	**0.244**	**0.273**	**0.293**	**0.282**
Outcomes	CS-Joint Complete	0.182	0.138	0.157	0.224	0.182	0.201
	CS-Joint +Marginals	**0.341**	**0.356**	**0.348**	**0.440**	**0.449**	**0.445**

Table: The performance of the CS-Joint model trained using only completely specified candidate triplets of the form $(c_P, c_{I/C}, c_O)$ versus a variant that accepts partially specified frames like $(\_, \_, c_O)$ or $(c_P, \_, c_O)$

Experimental Results

Category	Unseen concepts	Correctly classified
Population	193	24
Intervention	326	54
Outcome	423	77

Table: The number of unseen concepts identified correctly by the proposed CS-Joint model.

Conclusion

- We developed a new model for structured clinical text annotation that can work effectively with limited training data
- Our model learns to infer terms from the UMLS metathesaurus that describe the individual PICO elements relevant to a given study, as described in an input free-text
- It solves an important practical task for biomedical natural language processing.

Future Work

- Moving forward, we will further improve upon this model within the same framework, by exploiting the ontological structure underlying UMLS.
- We also hope to focus efforts on improving the recognition of novel terms, as this is important for the present task.

Acknowledgements

Thanks to SIGIR and NSF for grants towards attending CIKM'17. JT and GS acknowledge support from Cochrane via the Transform project. BCW was supported by the National Library of Medicine (NLM) of the National Institutes of Health (NIH), grant R01LM012086. IJM acknowledges support from the MRC (UK), through its grant MR/N015185/1.

Thank You!