Text Generation from Knowledge Graphs with Graph Transformers

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, Hannaneh Hajishirzi

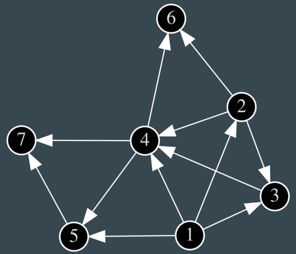
•••

Explanation and presentation
by
Pratik Karmakar

Objective:

Title of the paper

+



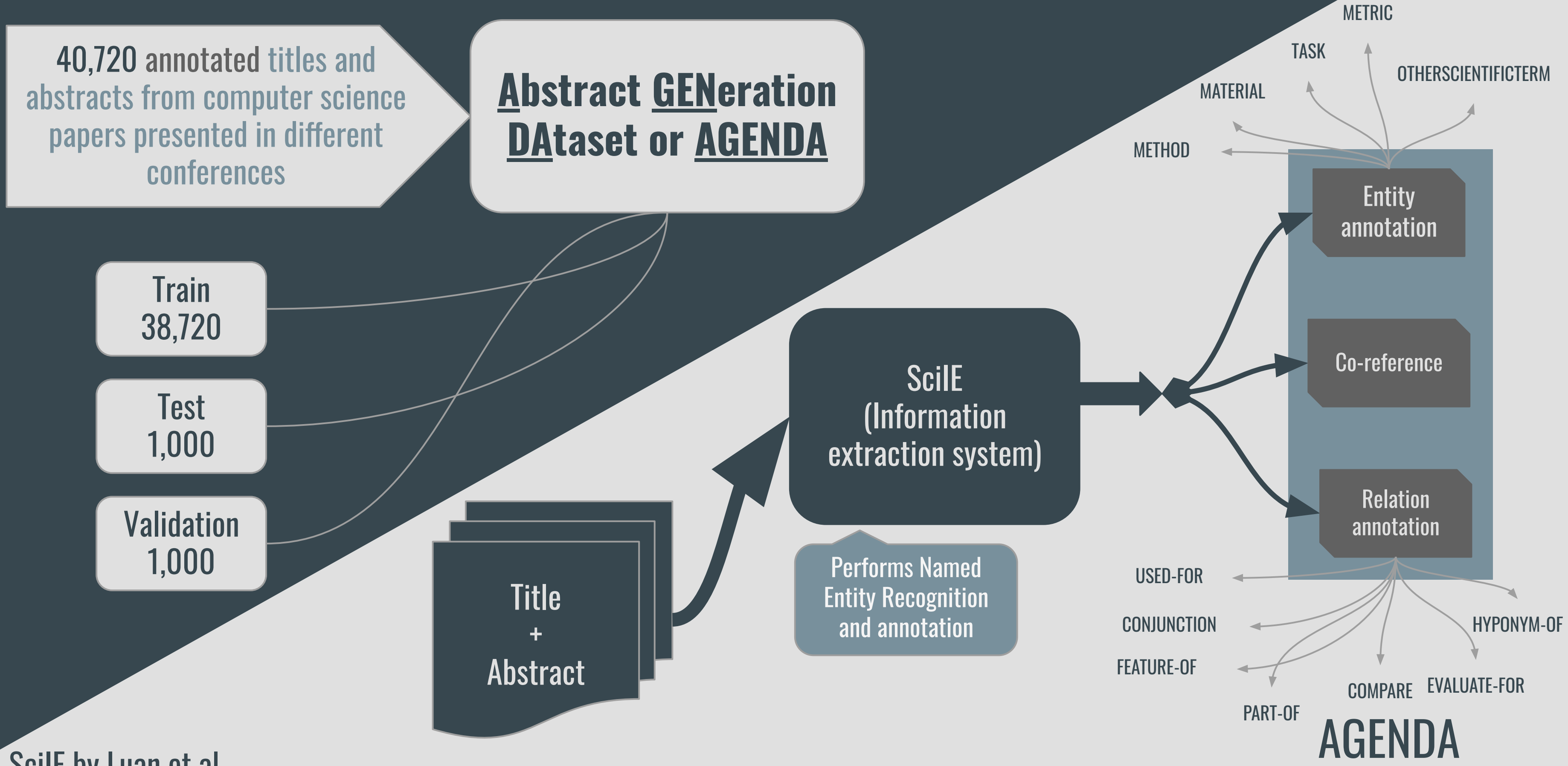
Knowledge graph

GraphWriter

The abstract of the paper:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco....

By Rik et al.

The Dataset:



SciIE by Luan et al.

Vocabulary
29k

Avg. length
9.9 words

Titles

Tokens
413k

AGENDA

Knowledge Graph

Avg. vertices
12.42

Avg. edges
4.43

Avg. length
141.2 words

Vocabulary
77k

Abstracts

Tokens
5.8M

AGENDA collected by Rik et al.

Data structure

Title	Entities	Entity types	Graph	Abstract (Target)

Entities: Semi-colon delimited list of entities (their position indices used in graph: head & tail in triplets)

Entity types: Space delimited list of entity types for each entity in "Entities" column

Graph: Semi-colon delimited list of graph triples: **<entity relation entity>**

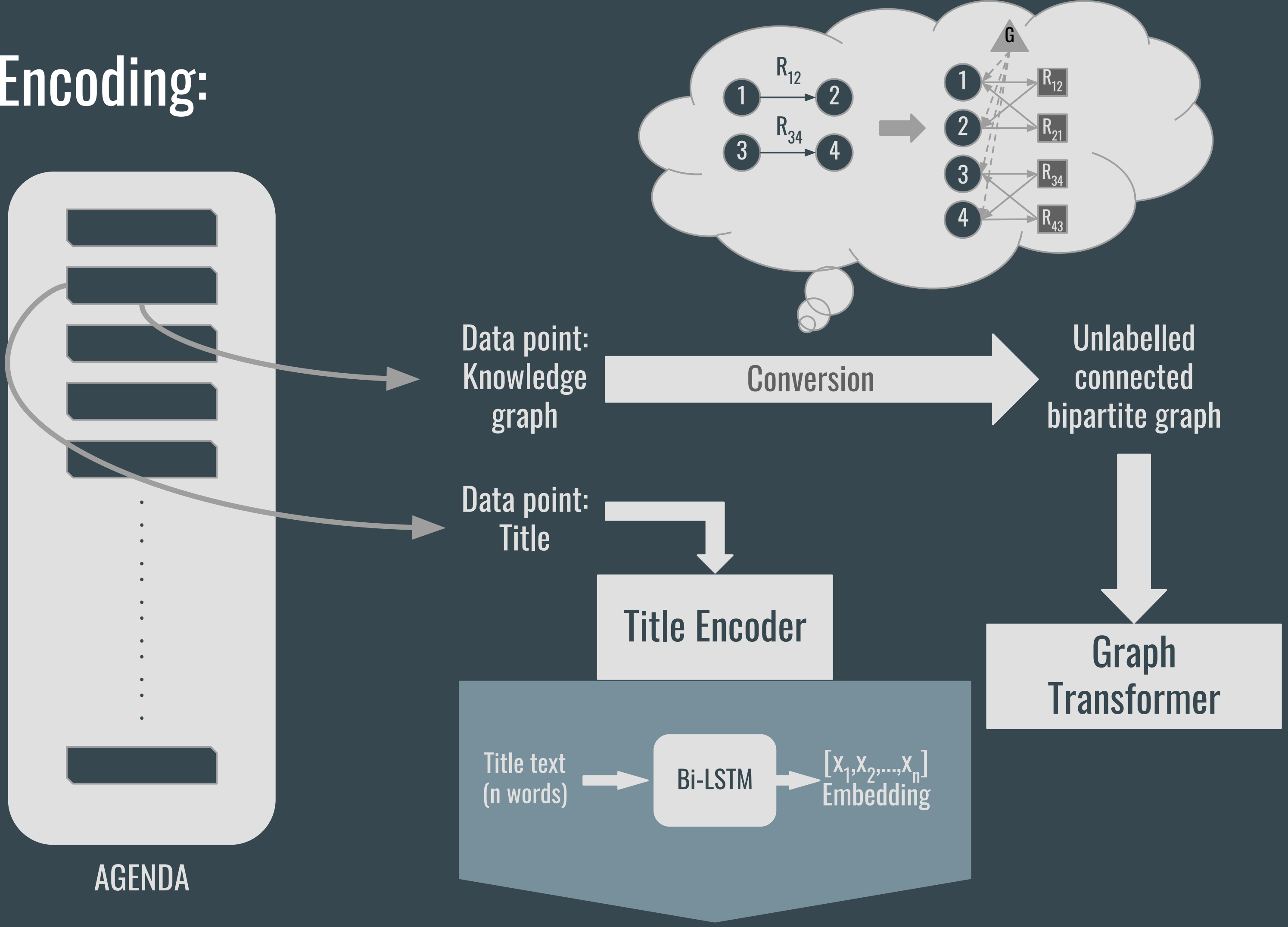
prior-free and prior-dependent regret bounds ; stochastic multi-armed bandit problem ; distribution-free and distribution-dependent bounds ; non-bayesian stochastic bandit ; thompson sampling ; bayesian regret ; prior distribution ; reward distributions

<otherscientificterm> <task> <otherscientificterm> <material> <method> <otherscientificterm> <otherscientificterm> <otherscientificterm>

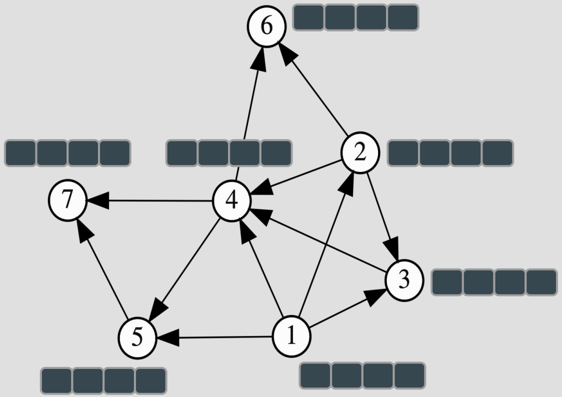
2 0 3 ; 7 2 1

distribution free and distribution-dependent bounds USED-FOR non-bayesian stochastic bandit; reward distribution FEATURE-OF stochastic multi-armed bandit problem

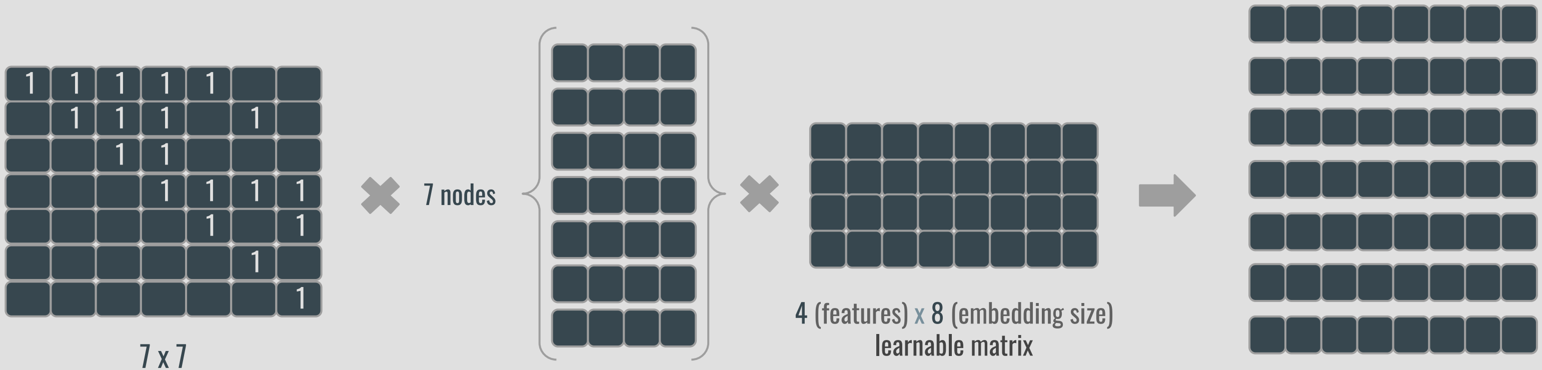
Encoding:


R12
1
2
R34
3
4
G
1
2
3
4
R12
R21
R34
R43
Data point:
Knowledge
graph
Conversion
Unlabelled
connected
bipartite graph
Data point:
Title
Title Encoder
Graph
Transformer
Title text
(n words)
Bi-LSTM
[x1,x2,...,xn]
Embedding
AGENDA

Graph Transformer



Graph
Feature vectors of nodes

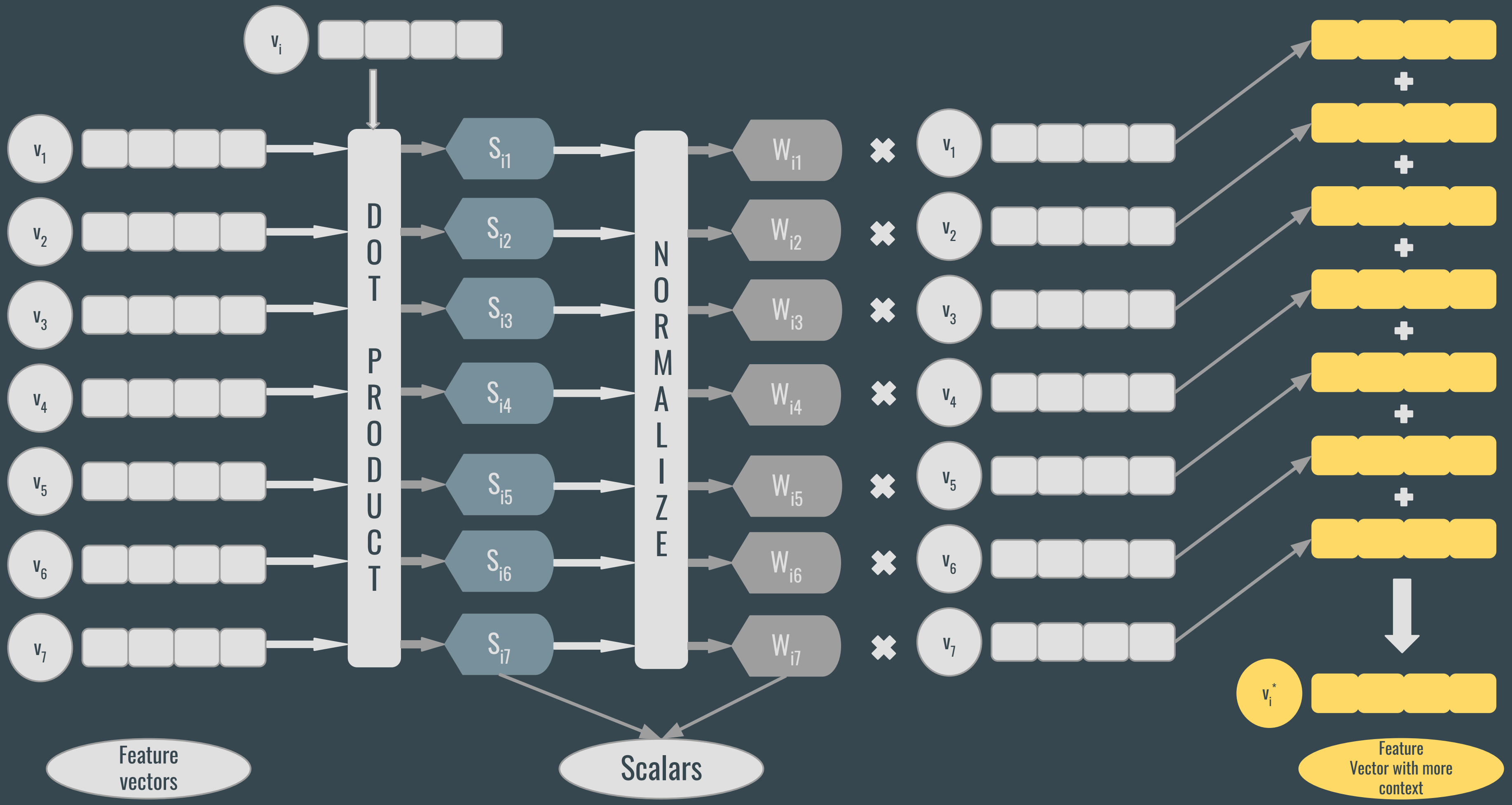


7 x 7
Adjacency matrix: Graph structure

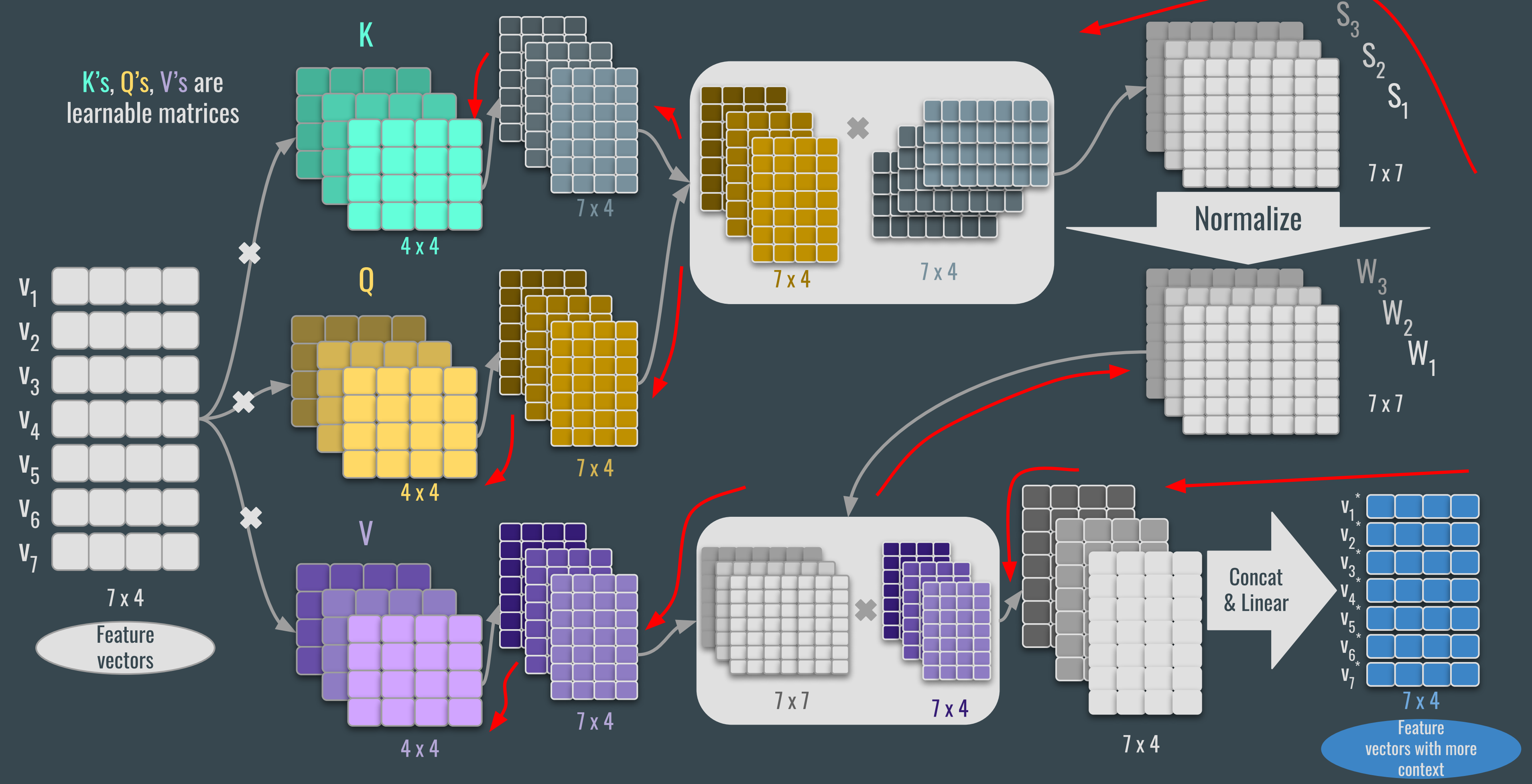
4
features

Node embeddings
Vectors of length 8
(here)

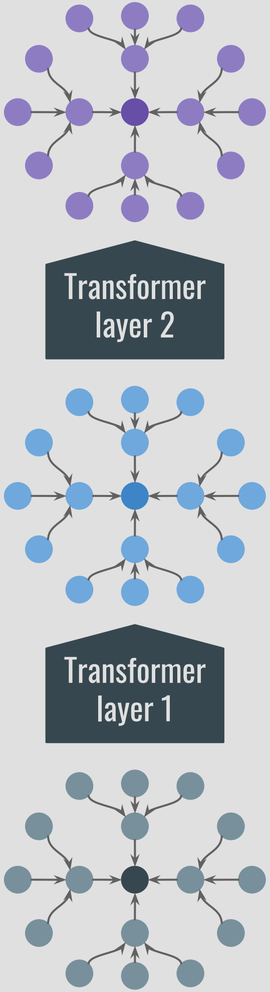
Self Attention Mechanism

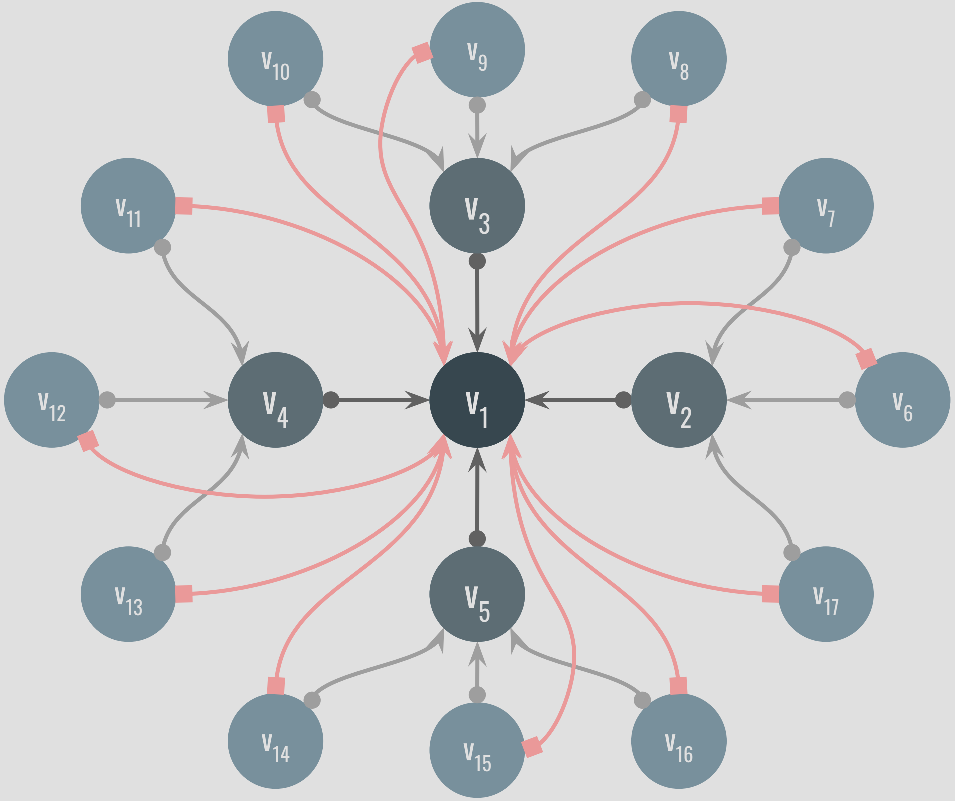


Multi-Headed Attention

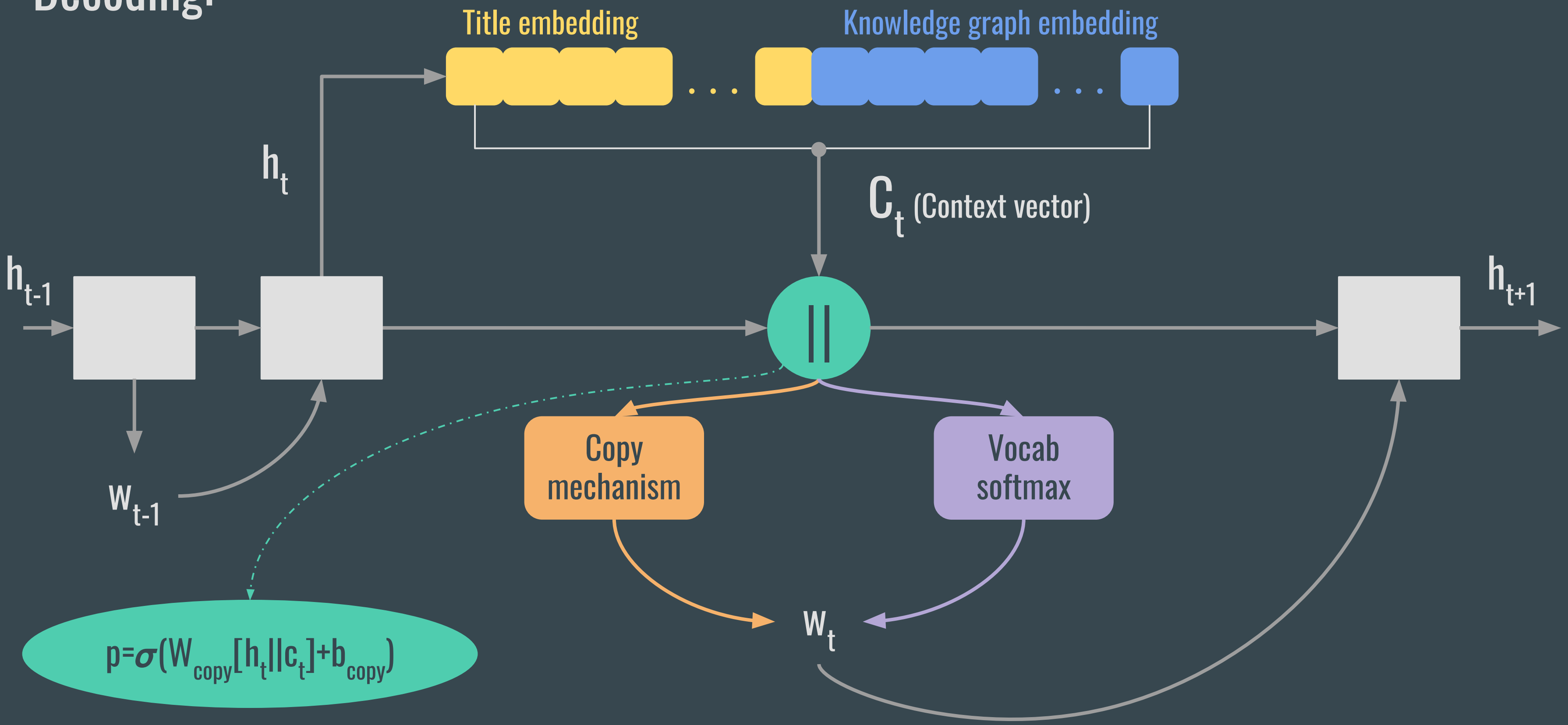


Contextualisation of the Knowledge Graph





Decoding:


Title embedding
Knowledge graph embedding
h_t
C_t (Context vector)
h_t-1
h_t+1
w_t-1
Copy mechanism
Vocab softmax
w_t
p=σ(W_copy[h_t||c_t]+b_copy)

Ground truth:

we present a LEARNING ARCHITECTURE for LEXICAL SEMANTIC CLASSIFICATION PROBLEMS that <unk> TASK-SPECIFIC TRAINING DATA with BACKGROUND DATA encoding general " world knowledge " . the LEARNING ARCHITECTURE compiles knowledge contained in a <unk> into additional training data , and integrates TASK-SPECIFIC AND BACKGROUND DATA through a novel HIERARCHICAL LEARNING ARCHITECTURE . experiments on a WORD SENSE DISAMBIGUATION TASK provide empirical evidence that this " HIERARCHICAL LEARNING ARCHITECTURE " outperforms a state-of-the-art standard " flat " one .

Input:

Title: Hierarchical Semantic Classification : Word Sense Disambiguation with World Knowledge.

Knowledge Graph: <learning architecture> <USED-FOR> <lexical semantic classification problems>

Model output:

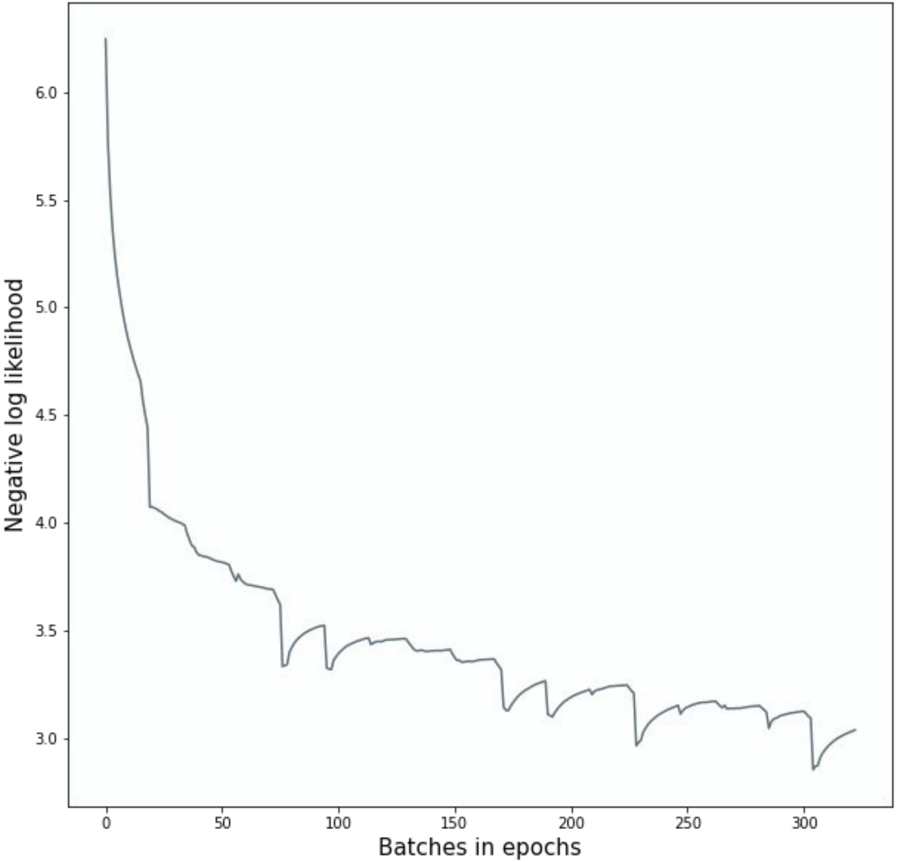
in this paper , we propose a LEARNING ARCHITECTURE for LEXICAL SEMANTIC CLASSIFICATION PROBLEMS . in particular , we focus on the task of LEXICAL SEMANTIC CLASSIFICATION PROBLEMS in the context of LEXICAL SEMANTIC CLASSIFICATION PROBLEMS . we propose a novel LEARNING ARCHITECTURE to tackle this problem . in particular , we propose a novel HIERARCHICAL LEARNING ARCHITECTURE to solve the LEXICAL SEMANTIC CLASSIFICATION PROBLEMS . we present a novel LEARNING ARCHITECTURE for LEXICAL SEMANTIC CLASSIFICATION PROBLEMS . we demonstrate the effectiveness of our LEARNING ARCHITECTURE on a WORD SENSE DISAMBIGUATION TASK .

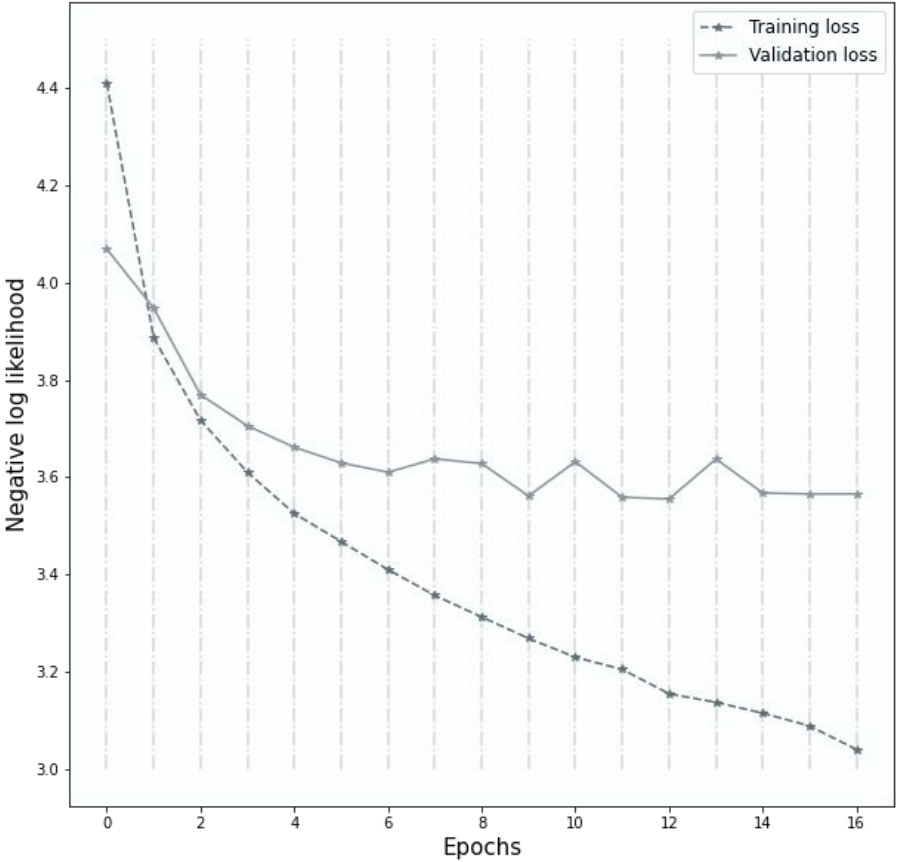
* Output after training the model for 17 epochs

Experimental setup:

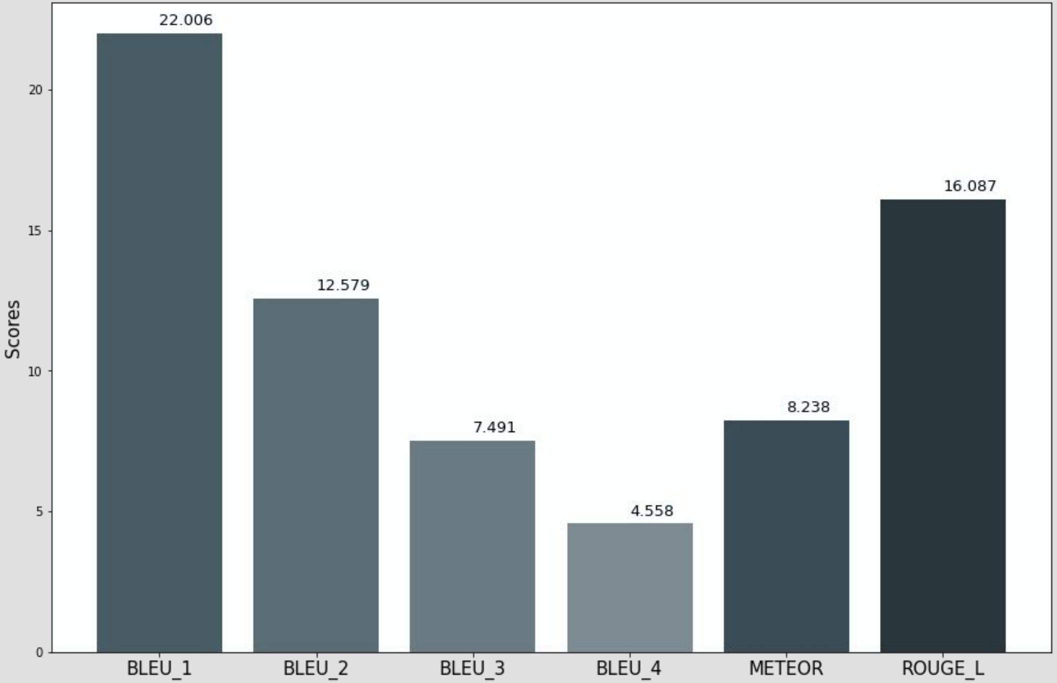
- Loss to minimize: Negative joint log likelihood of the target text and the generated entity
- Algorithm: SGD optimization with warm restarts (with early stopping based on validation loss)
- Dropout in self attention layers: 0.1
- Embedding dimension: 500
- Transformer layers: 6
- Number of attention heads: 4
- Word frequency threshold: 5 (a word below 5 is represented as <unk>)
- Decoding done with Beam Search (Beam size=4)
- Epochs: 17

Losses


6.0
5.5
5.0
4.5
4.0
3.5
3.0
Negative log likelihood
0
50
100
150
200
250
300
Batches in epochs


Training loss
Validation loss
4.4
4.2
4.0
3.8
3.6
3.4
3.2
3.0
Negative log likelihood
0
2
4
6
8
10
12
14
16
Epochs

Metric scores:



We have used 3 standard metrics for NLP:

1. **BLEU** (BiLingual Evaluation Understudy)

2. **METEOR** (Metric for Evaluation Of Translation with Explicit ORdering)

3. **ROUGE_L** (Recall Oriented Understudy for Gisting Evaluation (longest matching sequence of words))

All these metrics check for similarity between the reference text and generated text.

*The scores shown here are on the model trained for 17 epochs.

THANK YOU