Signal at TREC 2018 News Track

Dwane van der Sluis [*]
University College London and Signal
dwane.vandersluis@signal-ai.com

Dyaa Albakour
Signal
dyaa.albakour@signal-ai.com

Miguel Martinez
Signal
miguel.martinez@signal-ai.com

ABSTRACT

This paper provides an overview of the experiments we carried out for the entity ranking task at the TREC 2018 News Track. In particular, we experimented with adapting the supervised salience component of Salient Entity Linking (SEL), a state-of-the-art unified framework for entity linking and salience ranking. In our adaptation, we assume perfect entity linking performance and rank the entities using the salience components of SEL. Furthermore, in this adaptation, we aim to enhance the efficiency of the supervised salience ranking, and also to introduce sentiment-based features for entity salience.

1 INTRODUCTION

This paper presents our participation in the entity ranking task of the TREC 2018 News Track. The task involves ranking the list of mentioned entities given a news article according to how 'useful' they are to the reader to understand the article. In other words, the task aims at separating important entities from non-important ones within an article.

Indeed, the entity ranking task relates closely to 'entity salience', how related the entity is to the central discourse topic. Detecting and measuring entity salience is important for semantic search [3], knowledge extraction [6] and automatic summarisation [1]. Past approaches for measuring entity salience have included mining web query logs [2], utilising grammatical features [4] and graph based analysis, such as an adaptation of PageRank [5]. Recently Trani *et al.* [7] introduced Salient Entity Liking (SEL), a unified algorithm for entity linking (automatic annotation of text with entities in a Knowledge Base - KB) and salience ranking. The paper describing SEL [7] is one of the few publications in this area with a public dataset, and as far as we are aware, is the state-of-the-art for entity salience ranking.

In our participation at the news track, we make the assumption that entity salience ranking would reflect directly the entity ranking that should be achieved in the entity ranking task. Therefore, our participation evaluates entity salience ranking approaches that we develop by adapting the aforementioned SEL. SEL performs both entity linking and salience ranking using a unified supervised ranking algorithm. We adapt SEL to perform salience ranking while assuming a perfect entity linking performance. In other words, the linked entities are known in advance, and we use and adapt the salience component of SEL to rank these entities. In our adaptation, we focuses on two enhancements:

- develop a computationally efficient variant of SEL by reducing the number of expensive features required for the supervised ranking algorithm within SEL
- explore whether sentiment can be used to determine salience. In particular, we hypothesise that strong positive or negative sentiment expressed towards entities in a news article may indicate that it is salient or central to the main topic of the article. Therefore, we devise a set of features based on analysis of sentiment expressed towards specific entities in the article. We then use these features within the supervised salient ranking component of SEL

Following this, we submitted three different runs: one that reproduces SEL as closely as possible and two that reflect the enhancements explained above.

The structure of this paper is as follows. We present the adaptation of SEL and our enhancements in Section 2. Our experimental setup is described in Section 3 before introducing our runs in Section 4. Section 5 discusses our results. Finally we summarise our conclusions in Section 6.

2 ADAPTATION OF SEL

In this section, we give an overview of the SEL unified algorithm of entity linking and salience ranking and explain how we adapt it for the entity ranking task at the News Track.

In a nutshell, SEL works as follows. When performing entity linking, it generates a list of candidate entities from the reference KB (i.e. Wikipedia) and instead of making a binary decision on whether the entity is mentioned in the text, it assigns a salience score. A threshold on the salience score can be applied to perform entity linking. Also, the linked entities can be ranked by their salience score. Figure 1 depicts the pipeline of SEL. The pipeline consists of the following components:

(1) The spotter and candidate generator (Component A in Figure 1): Given a document D, this component identifies small portions of text, known as spots S_D in the document D. For each spot in S_D, the component generates a set of candidate entities that may be referred to by the spot. The union of all candidate entities are denoted with C_D.
(2) The light feature extractor (Component B in Figure 1): for each candidate entity in C_D, computationally cheap to calculate, 'light' features, are computed. These include features representing position, frequency and topographical characteristics of the entity.
(3) The binary classifier (Component C in Figure 1): trained on annotated documents, it receives the 'light' features as input and makes a decision to prune 'incorrect' candidate entities (not mentioned in the documents). It therefore

[*] This work has been done in the scope of a Master thesis in collaboration with Signal.

TREC'18, November 14-16, 2018, Gaithersburg, Md. USA.

DOI: 10.1145/nnnnnnn.nnnnnnn

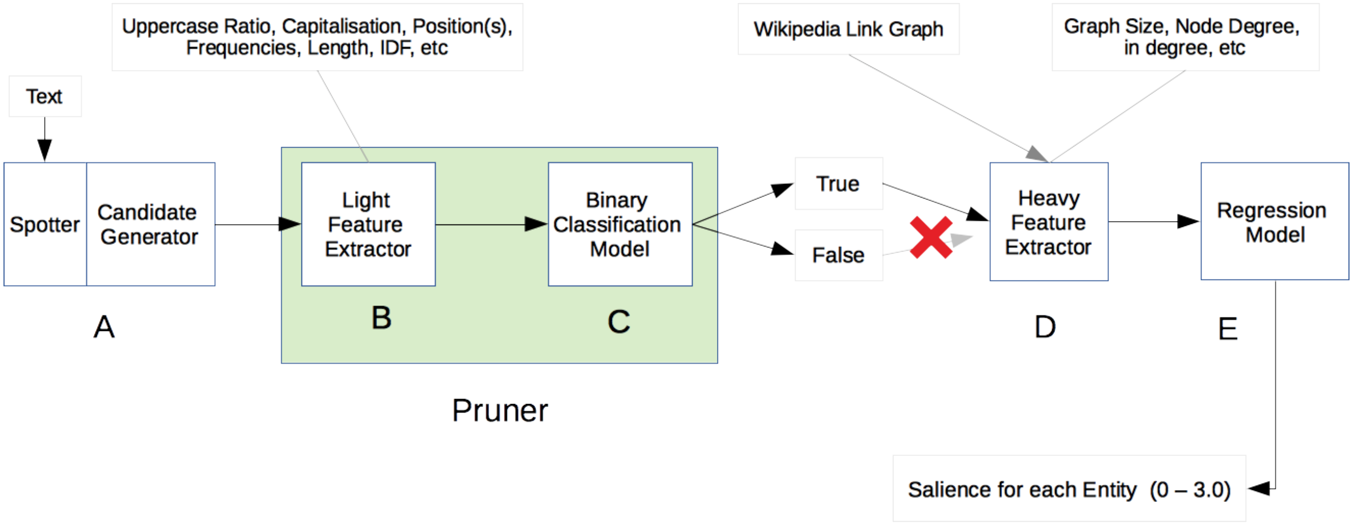
Uppercase Ratio, Capitalisation, Position(s), Frequencies, Length, IDF, etc
Wikipedia Link Graph
Graph Size, Node Degree, in degree, etc
Text
Spotter
Candidate Generator
Light Feature Extractor
Binary Classification Model
True
False
Heavy Feature Extractor
Regression Model
A
B
C
D
E
Pruner
Salience for each Entity (0 – 3.0)

Figure 1: Flow diagram of SEL as implemented by Trani [7]

helps to reduce the number of candidate entities for the following steps. 'Incorrect' candidate entities are filtered out from C_D to produce $C'_D \subset C_D$.

(4) The heavy feature extractor (Component D in Figure 1): first expands the list of pruned candidate entities C'_D to the set of entities contained in the KB graph that are at most 1 distant from C'_D (denoted with W_D) - a KB graph is a graph which models the relationships between entities in the KB. The 'heavy' features aim to model the relatedness of each candidate entity to other candidate entities within that graph. In particular, the 'heavy' features, are calculated for each pruned candidate entity in C'_D, conceptually by computing the centrality of the candidate entity to multiple sub-graphs of the KB graph created from W_D.

(5) The regression model (Component E in Figure 1): trained on documents annotated with the salience levels of the mentioned, i.e. linked, entities. It receives the full set of 'light' and 'heavy' features for each pruned candidate entity ce' in C'_D, and produces a salience score $sal(ce')$ for that entity. A threshold on the produced scores for pruned candidate entities can then be applied to produce the list of linked entities $L_D = \{e_1, e_2, ..\}$ (entity linking). Each entity in L_D can then be ranked by its salience score $sal(e)$ (saliency ranking)

In our adaptation of SEL, we focus on its salience-related components. We assume a perfect entity linking performance, i.e. the set of linked entities L_D are known in advance and the task is to assign a salience score for each entity in L_D. Therefore, the spotting, candidate generator, and candidate pruning are not needed, leaving extraction (components B and D) and the regression model (component E).

In the following subsection, we introduce two enhancements of SEL that we experimented with in our submissions:

2.1 Efficient Salient Features

One of the limitations of SEL is the complexity of computing the heavy features. Computing some of these features, involves traversing huge sub-graphs of the Wikipedia graph which may consist of hundreds of thousands of nodes. As a result, it is not feasible to use SEL for salient entity ranking in a practical real-time setting. Therefore, we developed an efficient variant of SEL. In particular, we conducted a feature analysis study, where we estimated the information gain of each light and heavy feature and its average running time. We then ranked the features by a function of their information gain and running time, favouring features with high information gain low running time. Following this, we selected top features from this ranking and use only those to train the regression model and predict salience scores.

2.2 Using Sentiment Analysis for Salience

News articles often express opinions or views about the central topic of the article. Therefore, we hypothesise that understanding the sentiment polarity around the entities mentioned in the document may indicate how salient they are to the topic of the article. To do that, we perform a dictionary-based sentiment analysis approach to devise additional features for each entity that aim to measure the sentiment polarity around it. In particular, applying a sliding window around the entity mention, we use a sentiment dictionary (in this case the AFINN 111 database) to assign a sentiment score for each word in the window. From this, we construct six different features to measure sentiment polarity as detailed in Table 1. We use these features in addition to the original SEL features (light and heavy) to train the regression model and predict salience scores.

3 EXPERIMENTAL SETUP

In this section, we detail the experimental setup for our submitted runs. In particular, we first focus on describing the datasets and resources required to train and to extract features for the adapted

Table 1: Sentiment-based Features for Salience

Title normalised summed sentiment of the 20 surrounding words
Title proportion of words that have negative sentiment in the title
Title proportion of words that have positive sentiment in the title
Body normalised summed sentiment of the 20 surrounding words
Body proportion of words that have negative sentiment in the body
Body proportion of words that have positive sentiment in the body

Table 2: Dexter dataset statistics. * HS Entities (Highly Salient) refer to entities with a salience score ≥ 2.0

Number of Articles	365
Number of Entities	5,460
Min Entities per Article	10
Max Entities per Article	32
Mean Entities per Article	14.9
Min Article Length	594
Max Article Length	2472
Mean Article Length	1599
Min HS Entities per Article *	0
Max HS Entities per Article *	10
Mean HS Entities per Article *	4.1
Min Salience	0.25
Max Salience	3.0
Mean Salience	1.56
Salience StdDev	0.56

SEL. Then, we describe some of the important implementation details.

3.1 Datasets

To train the supervised ranking component of SEL (the regression model), we used the **Dexter Entity Salience** dataset[1] used originally to train the SEL model [7]. It consists of 365 Wikinews articles published between November 2004 and June 2014. Each article contains the linked entities to the Wikipedia KB, as it is authored by Wikinews users, who can specify the Wikipedia entities in the article. The number of linked entities in these articles vary between 10 and 25 entities. Multiple Annotators assigned the salience of each linked entity within each article. Ground truth salience levels are detailed in Table 4. We also summarise the statistics of the dataset in Table 2.

In the News Track, the **Washington Post** dataset is used to source the topics for the entity ranking task (the articles for which the entities need to be ranked. To give the reader an idea of the similarities and differences between this dataset and the one we used for training our salience regression model, we summarise similar statistics in Table 3. Both datasets consist of news articles, in English, with a similar mean number of entities per article (14.9 vs. 16.5). However there were differences. Articles in the Dexter Entity Salience dataset (the training dataset) are far shorter in length than those in the Washington Post dataset (1,599 vs. 7,327 characters)

Table 3: Washington Post Dataset statistics.

Number of Articles	50
Min Entities per Article	3
Max Entities per Article	52
Mean Entities per Article	16.5
Min Article Length	1,855
Max Article Length	56,925
Mean Article Length	7,329

Table 4: Salience levels and descriptions in the Dexter Entity Salience Dataset [7]

	Name	Description
3	Top Relevant	The entity describes the main topics or the leading characters of a document
2	Highly Relevant	Satellite entities that are not necessary for understanding the document, but provide important facets
1	Partially Relevant	Entities that provide background information about the content of the document, but disregarding them would not affect negatively the comprehension of the document
0	Not Relevant	Any other entity that is not relevant or not mentioned in the article

3.2 Wikipedia Resources

Wikipedia is used as the KB reference of entities in the News Track. It is also the KB for entities in the Dexter training dataset. As in the original implementation of SEL, for the KB graph needed to calculate the light and heavy features, we use the Wikipedia link graph. The Wikipedia link graph is a graph where vertices are Wikipedia entities, and edges are the hyperlinks between Wikipedia pages representing these entities. To create this graph and the data structures necessary to calculate the features, we used a Wikipedia dump from June 2018 [2], and processed it using the packages provided by the Dexter developers. [3] [4]

3.3 Implementation Details

Each topic in the Entity Ranking task consists of a news article D from the Washington post and a list of mentioned entities in the article, where their Wikipedia identifiers (their names) are given. In other words, the set of linked entities L_D in each article (see Section 2) is given. However, the features used to train the supervised salience ranking also require knowledge of the spots S_D of these entities, i.e. where they are mentioned in the text (see Section 2). For this reason, we developed a 'soft-match mapper' to map the entities to their spots. The soft-match mapper takes the name of each entity and tries to find an exact match in the text (the title and the content of the article). If no matches are found,

[1] https://github.com/dexter/dexter-datasets/tree/master/entity-saliency

[2] http://ftp.acc.umu.se/mirror/wikimedia.org/dumps/enwiki/20180601/enwiki-20180601-pages-articles.xml.bz2

[3] https://github.com/dexter/elianto/blob/master/how-to-generate-a-wikinews-dataset.md

[4] https://github.com/diegoceccarelli/json-wikipedia

Table 5: Results of our runs

Run	**nDCG@5**	**P@5**
signal-ucl-sel	0.6071	0.6480
signal-ucl-eff	0.6084	0.6440
signal-ucl-slst	0.5772	0.6200
median	0.6153	0.6680

the exact match is relaxed by removing the last word of the entity name. This covers most of the entities, but for those not matched we assign a single random spot from the text of the document. Finally, to implement the regressor component of SEL, we use the sklearn implementation of the Gradient Boosting Regression Tree.

4 RUNS

We submitted three runs to the Entity Ranking task of the TREC news track. Each run is based on our described adaptation of SEL, but each has different sets of features:

- **signal-ucl-sel**: we use the majority of the light and heavy features implemented by the original SEL algorithm [7]. The full set of features used are the 'light' features listed in Table 6 and the 'heavy' features listed in Table 7.
- **signal-ucl-eff**: this run uses the enhancement of SEL for efficiency described in Section 2.1, where we selected a subset of the light and heavy features (26 from total 59) to improve efficiency while trying to maintain effectiveness.
- **signal-ucl-slst**: this run uses the enhancement of SEL described in Section 2.2, where we use sentiment-based features for salience, in addition to all the light and heavy features.

5 RESULTS

We report results of our runs in the entity ranking task in Table 5. We observe that the signal-ucl-eff run is of similar effectiveness to signal-ucl-sel. They both obtain very similar P@5 and nDCG@5, even though signal-ucl-eff uses far less features (26 vs 59). This validates our approach for efficient salience ranking - removing the features with low information gain and high complexity did not affect the effectiveness. The sentiment-based signal-ucl-slst run has lower P@5 and nDCG@5 than the two other runs. Adding the sentiment features was not effective as we originally hypothesised.

Overall, all our runs are on par with the median performance of all submitted runs to the track in terms of P@5 and nDCG@5. Indeed, document by document performance of the run signal-ucl-sel can be seen in Figure 2, in comparison to the median performance. It shows that this run is either on par or outperforms the median performance for about 80% of the topics (news articles), yet performs poorly for some topics.

6 CONCLUSION

For the entity ranking task at the TREC News track, we experimented with a supervised approach that measures entity salience to rank entities. To this end, we adapted the state-of-the-art SEL algorithm. Overall our adaptation is promising, as all our runs perform on par with the median of all submitted runs. Most notably, our enhancement for efficiency shows that the computational complexity of the SEL approach can be reduced with little or no performance loss. However, adding extra features capturing sentiment expressed towards entities degraded performance. We aim to investigate further by looking at more effective ways to measure sentiment as well as training the salience regressor model on different datasets.

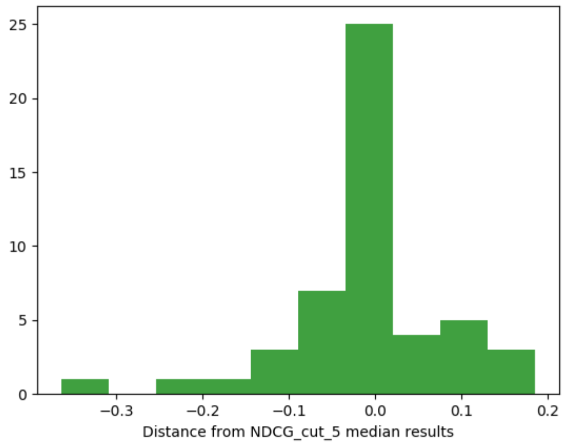

Figure 2: Distribution of the differences between nDCG@5 for the signal-ucl-sel run and the median nDCG@5.

REFERENCES

[1] Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics* 34, 1 (2008), 1–34.

[2] Bodo Billerbeck, Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, and Ralf Krestel. 2010. Ranking Entities Using Web Search Query Logs. In *Proceedings of the 14th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'10)*. Springer-Verlag, Berlin, Heidelberg, 273–281. http://dl.acm.org/citation.cfm?id=1887759.1887797

[3] Michael Gamon, Tae Yano, Xinying Song, Johnson Apacible, and Patrick Pantel. 2013. Identifying salient entities in web pages. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2375–2380.

[4] Eleni Miltsakaki. 2007. A rethink of the relationship between salience and anaphora resolution. In *Proceedings of the 6th discourse anaphora and anaphor resolution colloquium*.

[5] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[6] Dominic Seyler, Mohamed Yahya, and Klaus Berberich. 2017. Knowledge questions from knowledge graphs. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*. ACM, 11–18.

[7] Salvatore Trani, Claudio Lucchese, Raffaele Perego, David E. Losada, Diego Ceccarelli, and Salvatore Orlando. 2017. SEL: A unified algorithm for salient entity linking. *Computational Intelligence* 34, 1 (2017), 2–29. DOI: http://dx.doi.org/10.1111/coin.12147 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/coin.12147

Table 6: List of implemented light features from SEL. These features were used in all 3 runs. This is a subset of those implemented by SEL.

Feature
Min normalised position
Max normalised position
Mean normalised position
Standard deviation of normalised positions
Normed position in first 3 sentences in body
Normed position in middle sentences in body
Normed position in last 3 sentences in body
Normed position in title
Average normed position within sentences
Freq in first 3 sentences in body
Freq in middle sentences in body
Freq in last 3 sentences in body
Freq in title
True iff at least one mention is entirely upper-case
Maximum proportion of upper-case letters
Average character length of spots
Average term (word) length of spots
Is in title
Number of times the document refers to the entity
Spot ambiguity : $1.0 - (1.0/n)$ where n = num of candidate entities for the spot
In-degree of cj in the Wikipedia link graph
Out-degree degree of cj in the Wikipedia link graph
Undirected degree of cj in the Wikipedia link graph
Character length of document

Table 7: List of SEL heavy features implemented. This is a subset of those implemented by SEL.

Feature
Graph Size
Graph Diameter
Node Degree
Node average in degree
Node median in degree
Node average out degree
Node median out degree
Node average in-out degree
Node median in-out degree
Farness : Sum of the shortest paths between entity and all others
Closeness :1/ Farness
Eigan vector centrality
...
The above 12 features are then repeated 6 times, for 3 weights (edges evenly weighted, edges weighted by the Milne and Witten Relatedness score and weighted with Milne and Witten Relatedness with edges of 0 pruned) and 2 graph size variations (with and without adjacent nodes)