# トピック情報に注目した効率的な Salient Entity 検出手法

宮本 達朗† 北川 博之‡

†筑波大学大学院システム情報工学研究科 〒 305-8573 茨城県つくば市天王台 1-1-1
‡筑波大学計算科学研究センター 〒 305-8573 茨城県つくば市天王台 1-1-1
E-mail: †miyamoto@kde.cs.tsukuba.ac.jp, ‡kitagawa@cs.tsukuba.ac.jp

**あらまし** エンティティリンキングは自然言語文書に含まれるエンティティを示す語句であるメンションと，知識ベース中のエンティティを対応づける技術である．文書に含まれるエンティティの，文書内容に対する関連度はそれぞれ異なることが知られており，特に関連が大きいいくつかのエンティティは Salient Entity と呼ばれ，これらを検出することでより精密な文書検索や文書要約への応用が期待される．文書から Salient Entity を検出する既存手法の多くは，(1)エンティティリンキングにより文書中に含まれるエンティティを抽出し，(2)それらの文書内容への関連度を算出することで実現される．しかしながらこれらの手法には Salient Entity と紐付かないメンションに対しても正確なリンキングのための計算を行う必要があるという問題点がある．そこで本研究では，文書やメンション，メンションの候補エンティティ集合らのトピックを考慮することで，Salient Entity 検出のための効率的な候補エンティティの抽出を行うと同時に，Salient Entity と紐付くメンションの推定を行う手法を提案する．本稿では実データを用いた評価実験を行い，提案手法の有効性を検証する．

**キーワード** Salient Entity，エンティティリンキング，トピック

## 1. 序論

インターネットの発展と，それを通じての個人の情報発信が容易な時代となったことで，大量の自然言語テキストデータが日々生成・蓄積されている．自然言語テキストデータを機械が理解することで，これら大量のデータを機械的に整理・解析し，そこから有用な知見が得られる可能性がある．

文書の内容を機械的に理解するための手法の一つとしてエンティティリンキングがある．エンティティリンキングとは文書中のメンション(語句)に対して，知識ベース中の対応するエンティティを紐付ける技術である[1][2].

一般に文書には多数のメンションが含まれるためエンティティの数も多数ある．しかし，文書中のエンティティ全てが文書内容に関連がある訳ではなく，これらの中でも，特に密接な関連があると考えられるいくつかのエンティティは Salient Entity と呼ばれる．そして文書内容に対するエンティティの関連度は Saliency と呼ばれる．

例えば文書検索の場面では，エンティティリンキングによって文書中のエンティティが明らかになれば，「田中将大」に関する文書を検索する時，たとえ文書中に「田中」としか表記されていなくても「田中将大」に関する文書として検索を行うことができる．そしてさらにどのエンティティが文書の Salient Entity であるかの検出が行われていれば，「田中将大」に関する文書集合に対して，“「田中将大」のことをメインに取り扱った文書なのか“，という点で検索結果をランキングすることが可能である．

Salient Entity を検出するためのいくつかの手法[3][4][5][6]が提案されているが，これらの手法はいずれも，全てのメンションに適切なエンティティをリンキングさせた後に各エンティティの Saliency を推定しており，Salient Entity と紐付かないメンションに対しても正確なリンキングのための計算を行う必要があるほか，エンティティの Saliency を推定するために教師あり学習を用いているため，その訓練データセットを構築する手間がかかるという問題点がある．

そこで本研究では，文書やメンション，メンションの候補エンティティ集合らのトピックを考慮することで，Salient Entity 検出のための効率的な候補エンティティの枝刈りを行うと同時に，Salient Entity と紐づくメンションの推定を行い，それらのメンションのみに対してリンキングを行う手法を提案する．

本手法の基本的なアイデアは以下の通りである．

Salient Entity とは文書内容と密接に関連するエンティティであるため，文書と，Salient Entity やそれに紐付くメンションとは類似したトピックを持つと考え，文書とエンティティ間や，文書とメンション間のトピック類似度を利用してメンションと候補エンティティ集合の枝刈りを行う．

本稿では，Wikinews データセット[7]を用いた評価実験で提案手法の評価を行い，考察と検討を行った．

## 2. 関連研究

Salient Entity の検出手法は大きく 2 つに分けられる．1 つ目は文書中のエンティティに対して Saliency 推定を行い，Salient Entity を検出する手法[3][4][5]であり，2 つ目はエンティティリンキング と Salient Entity 検出を同時に行う Salient Entity Linking[6]である．前者

は Salient Entity 検出のみを目的とし，主に文書と文書中のエンティティとの関係性から Saliency 推定を行うのに対し，後者はエンティティリンキング と Salient Entity 検出を同時に行うことを目的としており，メンションとエンティティとの関係性も考慮して Saliency を推定し，メンションを正しいエンティティへリンクさせなくてはならない点が大きく異なる．

[3]では，Salient Entity 検出を，文書中のエンティティを Salient Entity か否かに分類する 2 値分類問題であると捉え，分類器が利用された．エンティティタイプ(人名, 地名, 組織名など)やエンティティの出現頻度，文書タイトルにエンティティが含まれているかどうかなどの情報を特徴量として分類器の学習を行うことにより，文書中のそれぞれのエンティティに対して Salient Entity か否かのラベル付けを行った．

[4]では，Web ページ上のテキストデータのみを対象に, Salient Entity を検出する手法が提案されている．この手法では，文書におけるエンティティの出現位置やエンティティの出現頻度, Web グラフ(Web ページのリンク構造)などの情報から特徴量を設計し，2 値分類器を学習させ，検出を行なっている．

[5]では，エンティティ出現頻度などの統計量の他に，文書の構文解析から得られる係り受け関係や，エンティティの分散表現などから特徴量を設計し，文書中のエンティティの Saliency を予測する回帰器を学習させ，Saliency が閾値より大きい文書中のエンティティを Salient Entity として検出を行なった．

これらの手法には，2 つの問題点があると考えられる．まず 1 つは，教師あり学習に用いるデータセット構築のコストの高さである．エンティティがどの程度文書の内容と関連しているかをスコアリングするためにはまず文書を理解しなければならないため機械的なスコアリングは困難だと考えられる. 手法[4]では, Web ページクリックログを利用することで機械的に学習データセット構築が行われているが，Web ページのみを対象としているため，汎用性という点で問題があると言える．手法[3][5]では，それぞれ複数人によるスコアリングの多数決，クラウドソーシングによって学習データセットの構築が行われており，構築コストが高いと言える．2 つ目の問題点は，Salient Entity 検出をエンティティリンキングの後続処理であると位置付けし，ひとまず全てのメンションに対してリンキングを行う点である．目的は Salient Entity の検出であるため，計算コストの面からみて，元々Salient Entity と対応付かないメンションに関してはリンキングのための計算を行わない方が好ましい．

Salient Entity Linking 手法[6]では，候補エンティティ集合を枝刈りした後に，統計的特徴量と，メンション・候補エンティティ・知識ベース中のリンク構造を用いて生成されるグラフから抽出される特徴量を組み合わせ，回帰器を学習させることで候補エンティティの Saliency の推定を行った．各メンションの候補エンティティのうち Saliency が最も高いエンティティをメンションとリンキングさせ，エンティティリンキングを行うとともに閾値以上の Saliency を持つエンティティを Salient Entity とした．

本研究は，エンティティの Saliency を推定するのではなく，文書とメンション，またエンティティ間のトピック類似度を用いてメンション集合と候補エンティティ集合を枝刈りし，既存リンキング手法と組み合わせることで効率的な Salient Entity 検出を行うことを目的としているためこれらの手法とは異なる．

## 3. 提案手法

本章では提案手法である，効率的に Salient Entity を検出するためのメンション集合および候補エンティティ集合に対する枝刈り手法について述べる．

3.1 節にて一般的なエンティティリンキングの概要を記すとともに提案手法のトピック情報に注目した Salient Entity 検出フレームワークについて述べる．

また，本手法ではメンション集合と候補エンティティ集合の枝刈りを行うために単語・エンティティ・文書の分散表現を用いており，3.2 節で分散表現の学習について述べ，学習した分散表現を活用したメンション集合と候補エンティティ集合の枝刈り手法について 3.3 節で述べる．3.4 節にて，リンキング手法について記す．

### 3.1 Salient Entity 検出フレームワーク

一般的にエンティティリンキングは，文書を入力として，(1)文書中からメンションとなり得る語句を抽出し，(2)抽出されたメンション集合のそれぞれに対して候補エンティティ集合を生成し，(3)各メンションに対して，その候補エンティティ集合の中から適切なエンティティを紐付ける，という 3 ステップによって実現され，メンションとそれと紐付くエンティティのペアのセットが出力される．

本研究では，上記のようなエンティティリンキングフレームワークに提案手法を組み込むことで効率的な Salient Entity の検出を行う．すなわちステップ(1)およびステップ(2)にて抽出，生成されたメンション集合および候補エンティティ集合を文書とのトピック類似度に基づいて枝刈りを行うことで，Salient Entity と紐付くメンションらにのみリンキングのための計算を行い，出力としては Salient Entity に紐付くメンションと正解 Salient Entity のペアのみのセットが期待される．トピ

ック類似度の計算にはあらかじめ学習された分散表現を用いており，3.2 節にてその詳細ついて述べる．

### 3.2 分散表現

一般にエンティティリンキングにおけるリンキングステップでは，メンションと候補エンティティとの関連度を考慮するほかに，同一文書内のエンティティは一貫性を持つという仮定から，ほかのメンションらの候補エンティティ同士との関連性を考慮することでより精度の高いリンキングを行っている．この際，これら関連度をいかに推定するかが重要になってくるが，本研究では，近年いくつかの手法[8][9]で採用されている分散表現学習を利用した手法を用いる．分散表現学習は，類似した語がベクトル空間上で近くに存在するように，単語を低次元ベクトル空間上に埋め込む技術である．

Skip-gram model[10]は著名な分散表現学習手法の 1 つであり，意味的に近い単語同士がベクトル空間で近く存在するような単語分散表現空間を学習する．

SCDV[11]は単語分散表現空間から，単語それぞれが持つトピックを考慮した別の単語分散表現空間を生成し，それを元に文書のスパースな分散表現を生成する手法である．すなわち生成された分散表現空間では，近しいトピックの分布を持つ単語や文書同士が，ベクトル空間上で近く存在する．

本研究では，単語とエンティティを同一空間に埋め込むために，Wikipedia 記事中のアンカーテキストの直前または直後に，そのリンク先のエンティティ(Wikipedia 記事)のタイトルを挿入するという処理によって構築した学習コーパスに対して Skip-gram model を用いることで，単語・エンティティ分散表現(Skip-gram 分散表現)を学習する．また，Skip-gram 分散表現に対して SCDV を適用しトピックを考慮した単語・エンティティ・文書分散表現(SCDV 分散表現)に拡張する．3.3 節にて，学習された分散表現を活用した枝刈り手法について述べる．

### 3.3 トピック類似度に基づく枝刈り

本節では Salient Entity 検出のための，トピック類似度に基づいたメンション集合および候補エンティティ集合に対する枝刈り手法について述べる．なお，トピック類似度の推定には 3.2 節にて述べた SCDV 分散表現を活用し，分散表現空間におけるベクトルの近さの尺度としては cos 類似度を用いた．

#### 3.3.1 メンション集合の枝刈り

1 章で述べた基本アイデアに基づき，Salient Entity と紐付くメンションは他のメンションと比べ文書とのトピック類似度が高いと考えられるため，メンション集合を文書とのトピック類似度でランキングした時のランク上位のメンションのみに対してリンキングを行う．

$$\boldsymbol{Topic\ Sim_{m_i}} = \boldsymbol{cos}\left(\overrightarrow{\boldsymbol{V_d}}, \overrightarrow{\boldsymbol{V_{m_i}}}\right) \qquad (1)$$

ただし$\vec{V}$は SCDV 分散表現を表す．

#### 3.3.2 候補エンティティ集合の枝刈り

候補エンティティ集合を枝刈りするにあたって，リンキングステップにおいて正確なリンキングが達成されるために必要な条件として以下の 2 点が考えられる．

**① メンションの正解エンティティ**
**② 文書とのトピック類似度が高いエンティティ**

しかしながら本手法における候補エンティティ集合の枝刈りの目的はリンキングステップにおける，Salient Entity に対する正確なリンキングであるため，文書とのトピック類似度が高いエンティティを残すように枝刈りを行うことで両条件を満たす枝刈りの実現が期待できる．

3.3.1 節と同様，全てのメンションの候補エンティティを集約したものを文書とのトピック類似度でランキングし，ランク上位のエンティティで新たな候補エンティティ集合を生成し，これらのエンティティのみを対象にリンキングを行う．

$$\boldsymbol{Topic\ Sim_{e_j}} = \boldsymbol{cos}\left(\overrightarrow{\boldsymbol{V_d}}, \overrightarrow{\boldsymbol{V_{e_j}}}\right) \qquad (2)$$

また，この段階で候補エンティティが全て枝刈りされたメンションは Salient Entity とは紐付かないメンションであるとみなし，リンキング処理の対象に含まないものとする．

### 3.4 リンキングステップ

本節では Salient Entity 検出フレームワークに用いるリンキング手法について述べる．本研究では，提案手法を組み込むリンキング手法として Phan ら[8]が提案した Pair-Linking を用いた．Pair-Linking ではメンションがある候補エンティティと紐付く確信度を，ほかのメンションとその候補エンティティとの組み合わせの内，最も確信度が高い組み合わせを基に推定することで，全ての組み合わせを考慮せずに精度を保ちつつ高速にリンキングを行う手法である．

本手法において，リンキングステップの結果として出力されるのはメンションと Salient Entity のペアのセットであるため，リンキング処理の結果が Salient

Entity 検出の結果となる.

メンションとエンティティ間や，エンティティ同士の関連度の推定は，3.2 節で述べた Skip-gram 分散表現を活用し，分散表現空間におけるベクトルの近さの尺度としては cos 類似度を用いた.

## 4. 検証実験

3 章で提案した Salient Entity 検出手法の検出精度を評価するために, Trani ら[6]が作成した Wikinews データセット[7]を用いた評価実験を行った．本章では 4.1 節で実験設定を，4.2 節で実験結果を示し，4.3 節で考察を記す.

### 4.1 実験設定

本実験では提案手法の有効性を検証するため，メンション集合および候補エンティティ集合を枝刈りする際の閾値を変化させたときの Salient Entity 検出精度を評価する． また, 枝刈り後の候補エンティティ集合に対しては，Salient Entity を含んでいるか否かという点での評価を行う．したがって評価指標としては recall, precision, f1-score を用いた.

なお，本実験においては Salient Entity 検出フレームワークへの入力として {文書$d$, メンション集合$M$, 候補エンティティ集合$C_d$} が与えられるものとする.

#### 4.1.1 実験データセット

本実験では，実験データセットとして Trani ら[6]が作成した Wikinews データセット[7]を用いた．Wikinews データセットは 365 件のニュース文書から構成されており，それぞれの文書には，文書中のエンティティのリストが紐づいており，それぞれのエンティティには Saliency スコアが与えられている. Saliency スコアはクラウドソーシングを用いて，複数人が{0, 1, 2, 3}の 4 段階(3 が最も文書との関連度が高い)で評価を行い，その平均をとったものである．また，本実験では Saliency スコアが 2 以上のエンティティを Salient Entity であると定義した.

Wikinews データセットでは, 文書のどの語句がメンションであるかが関連づけられていないため，本実験では人手で文書中のエンティティと紐付くと考えられるメンションを抽出し実験を行った.

実験データセットの詳細を表 1 に示す.

**表 1 実験データセット**

| 文書数 | 平均エンティティ数 | 全エンティティ数 | Salient Entity |
|---|---|---|---|
| 365 | 12.9 | 4715 | 1320 |

**表 2 辞書に基づく候補エンティティ集合**

| 候補エンティティ総数 | 613284 |
|---|---|
| 1 メンションあたりのエンティティ数 | 130 |
| Recall | 0.949 |
| Precision | 0.0143 |
| F1-score | 0.0282 |

#### 4.1.2 事前処理

実験を行うにあたって，事前処理として分散表現の学習と候補エンティティ集合を生成するための辞書の構築を行った.

**A) 分散表現学習**

分散表現学習の学習コーパスとしては Wikipedia 公式が配布している英語版 Wikipedia ダンプデータ[12]を用いた．本実験では 2018 年 6 月 7 日時点での Wikipedia 記事データを利用した.

また，Skip-gram model のパラメータは次元数 200, window size 5, minimum count 5 として Skip-gram 分散表現の学習を行った．SCDV のパラメータはトピック数 100, Sparse parameter0.04 として SCDV 分散表現への拡張を行った.

Skip-gram model での分散表現学習には Python の gensim ライブラリを利用した.

**B) 辞書の構築**

多くの手法では候補エンティティ集合の生成には事前に構築した辞書が利用されている[2]. 本実験では一般的な辞書構築手法である Wikipedia を利用した手法で辞書を構築した. Wikipedia の記事をエンティティとし，アンカーテキスト構造を利用して，アンカーテキスト文字列(メンション)とリンク先 Wikipedia 記事(エンティティ)のペアを抽出した後，メンションに関して集約することで辞書の構築を行った.

構築した辞書から実験データセットのメンションの候補エンティティ集合を生成した場合，どの程度正解エンティティを含むのかを表 2 に示す.

### 4.2 実験結果

#### 4.2.1 メンション集合の枝刈り

本研究においてメンションの枝刈りは，文書とメンションのトピック類似度によってメンションをランキングし，Top-K のトピック類似度を持つメンションのみを残す方法で行った. 結果を図 1 に示す. X 軸が K,

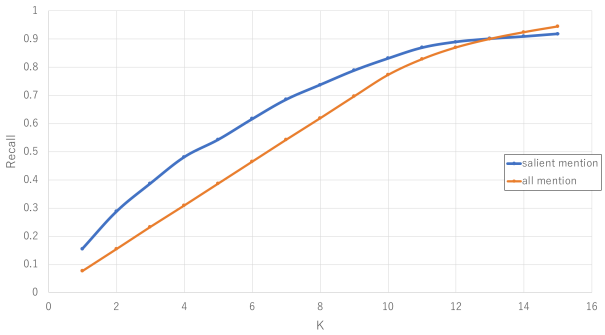


図1 メンション集合の枝刈り

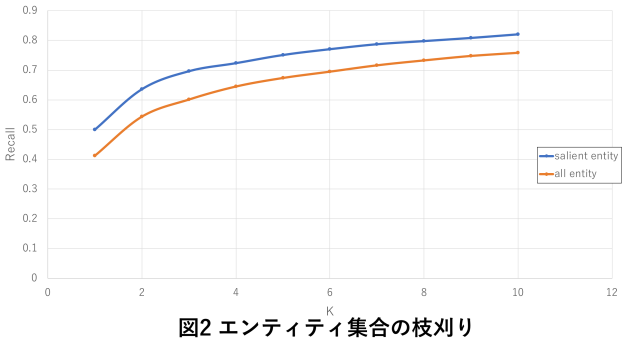


図2 エンティティ集合の枝刈り

Y 軸がメンションに対する Recall の文書平均を示す．

K を変化させた時，Salient Entity と紐付くメンション(Salient mention)とその他のメンションを Recall で比較してみると，Salient mention の方が，そのほかのメンションよりもより多く枝刈りで残る傾向が見られた．したがってトピック類似度で Salient mention をより多く残すような枝刈りすることがある程度は可能ではないかと思われる．

### 4.2.2 候補エンティティ集合の枝刈り

本研究において候補エンティティの枝刈りは，4.2.1 節同様，文書とエンティティのトピック類似度によるランキングを行った後，Top-K のトピック類似度を持つエンティティのみを残す方法で行った．結果を図 2 に示す．X 軸が K，Y 軸がエンティティに対する Recall の文書平均を示す．

メンション集合の枝刈りでの結果と同様に，Salient Entity はそのほかのエンティティよりもトピック類似度での枝刈りで残りやすい傾向を確認できた．

### 4.2.3 Salient Entity 検出

Salient Entity 検出の精度を検証するため，メンション集合と候補エンティティ集合の両方を枝刈りし，枝刈り後のメンションに対してリンキング処理をする実験を行なった．

メンション集合に関してはトピック類似度 Top-K で枝刈りを行い，K の変化による Salient Entity 検出精度の変化を検証した．候補エンティティ集合に関しては Recall 高く保つためにトピック類似度 Top-10 で枝刈りすると固定して実験を行なった．結果を表 3 に示す．実験データセット中には 1320 個の Salient Entity が含まれているが，メンションを Top-6 で枝刈りした場合に最も f1-score が高い結果となった．しかしながら K をこれ以上変化させても Recall は高くなる一方で Precision は低くなることが予想され f1-score は頭打ちになると思われる．

表3 Salient Entity検出　結果

| メンション　top-k | recall | precision | f1_score | リンキング 結果 (salient entityの正解数) | 枝刈り後のメンション |
|---|---|---|---|---|---|
| 1 | 0.0348 | 0.133 | 0.0552 | 46 | 345 |
| 2 | 0.206 | 0.372 | 0.265 | 272 | 730 |
| 3 | 0.284 | 0.342 | 0.310 | 375 | 1096 |
| 4 | 0.362 | 0.327 | 0.343 | 478 | 1460 |
| 5 | 0.408 | 0.295 | 0.342 | 539 | 1825 |
| 6 | 0.458 | 0.276 | 0.344 | 605 | 2190 |

### 4.3 考察

提案手法によってメンション集合および候補エンティティ集合を文書とのトピック類似度で枝刈りした結果，Salient Entity や，それに紐付くメンションは他のエンティティおよびメンションらに比べより多く残る傾向が見られた．しかしながらまだ十分な精度で枝刈りを行うことができているとは言えないと考えられ，その要因の 1 つとして SCDV で分散表現を拡張する際にエンティティのトピックをうまく捉えられていないのではないかということが考えられる．

候補エンティティ集合の枝刈りの際に閾値を設定し，トピック類似度がそれ以下のエンティティを枝刈りする方法で候補エンティティ集合を枝刈りした結果を図 3 に示す．

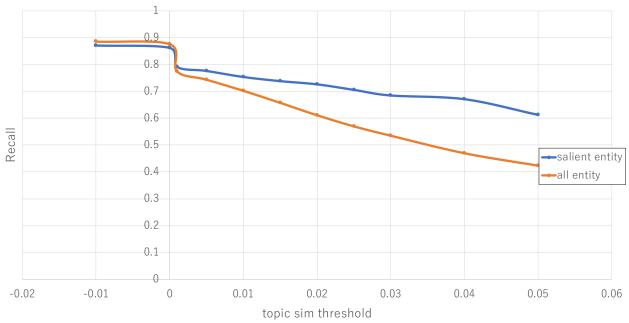


図3 エンティティ集合の枝刈り(topic sim threshold)

トピック類似度に対する閾値で枝刈りした場合，0 を超えたところで急激にエンティティの Recall が下がっており，これはエンティティがトピックを正確に捉えきれていないことを示すのではないかと考えられる．

Salient Entity 検出の結果に関しては，いかに効率的なメンション集合および候補エンティティ集合の枝刈りを行うかが大きく影響を与えると考えられるが，現状の結果をみると，それほど良い結果とは言えないと

考えるため，メンションやエンティティ，文書のトピックを適切に捉えられるようなSCDVのパラメータを与える必要がある．

また，一般にリンキングステップではメンションとエンティティの関連度に加え，同一文書に出現するエンティティは一貫しているとの仮定から，エンティティ同士の関連度も考慮することでより正確なリンキングを実現している．そのためメンションを枝刈りすることでリンキング自体の精度に影響を与えてしまうことが起こりうる．したがって，今後，枝刈りによるリンキングステップへの影響も調査する必要があると考えている．

## 5. まとめ

本研究では，効率的なSalient Entityの検出を目的とし，メンションやエンティティそして文書の，トピックに注目することでSalient Entityと紐付かないと考えられるメンションやSalient Entityになり得ないエンティティを枝刈りすることで，最も計算コストの高いリンキングステップをできるだけ効率的に行う手法を提案した．実データを用いた評価実験で，提案した枝刈り手法が有効に作用することを示唆した．

今後の課題としては，(1)適切なトピック数パラメータの分析，(2)枝刈りによるリンキング精度への影響の調査が挙げられる．

## 謝　辞

本研究の一部は，平成30年度共同研究(SKY株式会社) (CPE30034) 「データエンジニアリングの知見の応用によるSKYSEA Client Viewのログおよび資産情報の処理の高速化・軽量化・高度化」の助成を受けたものです．

## 参　考　文　献

[1] R.Mihalcea and A. Csomai, Wikify!: linking documents to encyclopedic knowledge. in Proc CIKM, pp233-242, 2007.

[2] W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base. Issues, techniques, and solutions. IEEE Transactions on Knowledge and Data Engineering, Vol. 27, pp. 443-460, 2015.

[3] L. Zhang, Y. Pan, T. Zhang, Focused Named Entity Recognition using Machine Learning, in Proc SIGIR, pp281-288, 2004

[4] M. Gamon, T. Yano, X. Song, J. Apacible, P Pantel, Identifying salient entities in web pages. In Proc CIKM, pp2375-2380, 2013.

[5] M. Ponza, P. Ferragina and F. Piccinno, SWAT: A System for Detectiong Salient Wikipedia Entitits in Texts, in CoRR, 2018.

[6] S. Trani, C. Lucchese, D. E. Losada, D. Ceccarelli, SEL: A unified algorithm for salient entity linking, in Computational Intelligence, 2017.

[7] https://github.com/dexter/dexter-datasets

[8] M. C. Phan, A. Sun, Y. Tay, J. Han, and C. Li, Neupl: Attentionbased semantic matching and pair-linking for entity disambiguation, in Proc CIKM, 2017.

[9] I, Yamada, H. Shindo, H. Takeda, Y. Takefuji, Joint learning of the embedding of words and entities for named entity disambiguation, in Proc CoNLL, pp250-259, 2016.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in Proc ICML, pp1188-1196, 2014.

[11] D. Mekala, V. Gupta, B. Paranjape, H. Karnick, SCDV: Sparse Composite Document Vectors using soft clustering over distributional representations, in Proc EMNLP, pp659-669, 2017.

[12] https://dumps.wikimedia.org/enwiki/latest/