

文の内容性と接続性を目的関数とする複数の評価文書の要約

西川 仁 長谷川 隆明 松尾 義博 菊井 玄一郎

日本電信電話株式会社 NTT サイバースペース研究所

{nishikawa.hitoshi,hasegawa.takaaki,matsuo.yoshihiro,kikui.genichiro}@lab.ntt.co.jp

1 はじめに

飲食店のレビューなど、商品やサービスに対する個人の評価が記述された評価文書がウェブ上には大量に蓄積されている。これらは消費者の立場からは購買行動に利用でき、企業の立場からは新商品の開発など企業活動の意思決定に利用できる。これらの評価文書の規模は大きく、効率的なアクセス手段の開発が不可欠である。その1つとして、テキスト自動要約技術は有望である。

近年のテキスト自動要約は、文書を文などの単位に分割した上で、文の集合に対して何らかのスコア関数を定義し、スコアが高い文(以下、重要文)の組み合わせを選び出すことによって行われることが多い[6, 13]。これらの手法は要約の対象とする文書の内容を可能な限り要約に含めていると考えられるが、読みやすさが考慮されていないという問題がある。

要約を読みやすくするための一つの工夫として、選択された文の集合に対して並びを与えるのは自然なアイデアである[2, 10]。しかしながら、文を選択した後に並びを与える際、どのように並べても自然に繋がり得ない文が選択されている場合、自然な要約を生成することが難しい。

そこで、本稿では、文を選んでから並べるのではなく、文の順列を直接探索し要約を生成する手法を提案する。我々は要約を重要文の順列とみなし、文の順列に対し内容に関するスコア(以下、内容性スコア)と文の並びの自然さに関するスコア(以下、接続性スコア)を定義し、これらのスコアの和が高い順列を要約とする。これにより、内容の観点からも、読みやすさの観点からも優れた要約が生成されることを評価実験によって示す。

以下、2節では関連研究について述べ、3節では要約を生成する具体的な手法について述べる。4節では評価実験の結果を報告し、5節でまとめる。

2 関連研究

複数の評価文書の要約[8, 4, 11]は、基本的には、まず要約の対象とする評価文書の集合に対し情報抽出を行い、評価対象に対しどのような観点(以下、評価属性)でどのような評価(以下、評価極性)が行われているかを特定する。その上でそれらの情報を用い重要文を抽出し、要約を生成する。出力となる要約の形態は、文のリスト[8]と、ひとまとまりのテキスト[4, 11]に大別することができる。本稿ではひとまとまりのテキストを出力とすることを目的とする。[4, 11]の問題は、要約を生成する際に内容しか考慮していないことである。ひとまとまりのテキストを出力とする場合には、読みやすさも考慮する必要がある。

要約を読みやすくするため、文の集合に並びを与えるものとしては、文書が投稿された日時を利用するもの[2]、コーパスを用いて文の接続確率を計算するもの[10]があるが、いずれも予め重要文として選択された文の集合に並びを与えるものである。そのため前述したように他の文と自然に繋がり得ない文が選択されている場合には自然な要約を生成することが難しい。

本稿では、既に選択されている文の集合に対して並びを与えるのではなく、要約対象の文書集合から直接文の順列を探索することにより要約を得る。我々の知る限り、複数文書要約において内容性と接続性を同時に考慮した研究はこれまでにない。

3 重要文の順列の探索

3.1 全体の構成

本稿では、文の順列に対し内容の良さを表わす内容性スコアと文の並びの自然さを表わす接続性スコアを定義し、定められた要約長の下でこれらの和が最大の順列を要約とする。形式的には、 s_0 と s_{n+1} をそれぞれ文の順列の先頭と末尾を表わす記号として、 n 個の文の順列である要約 $S^* = s_0, s_1, \dots, s_n, s_{n+1}$ を以下のように定義する。

$$S^* = \operatorname{argmax}_{S \subseteq T} [\text{Content}(S) + \lambda \text{Connect}(S)] \quad (1)$$

s.t. $\text{length}(S) \leq K$

ここで、 $\text{Content}(S)$ は文の順列 S の内容性スコア、 $\text{Connect}(S)$ は接続性スコア、 T は要約の対象とする文書集合から構成可能な文の順列の全て、 λ は内容性スコアと接続性スコアの重みを調整するパラメータ、 $\text{length}(S)$ は S の長さ、 K は要約のサイズの上限である。

以下、内容性スコアと接続性スコア、及び順列の探索について述べる。

3.2 内容性スコア

本稿では評価文書を要約の対象とするため、内容性スコア $\text{Content}(S)$ を評価情報 e を通じて以下のように定義する。

$$\text{Content}(S) = \sum_{e \in E(S)} f(e) \quad (2)$$

評価情報 e は評価属性 a と評価極性 $p = \{-1, 0, +1\}$ の組 $e = \langle a, p \rangle$ として定義^{*1}し、 $E(S)$ は S に含まれる評価情報の集合、 $f(e)$ は評価情報 e のスコアとする。すなわち、要約の内容性スコアは、文の順列 S が含む評価情報のスコアの和とする。ただし、同一の評価情報が2つ以上 S に含まれていたとしても、

^{*1} 評価対象は抽出しない。今回は特定の評価対象に紐付いた文書を要約の対象とするため、抽出される評価情報は全て特定の評価対象に対するものであるとする。

その評価情報に関しては1度しか内容性スコアに加算しないことにする。これにより、同一の情報が複数回現れるような冗長な要約が生成されづらくなる。今回、 $f(e)$ は要約の対象とする文書集合中において評価情報 e が現れる回数とした。

矛盾した評価情報が要約に含まれることを避けるため、評価属性が同一の評価情報のうち、評価極性が -1 のものと $+1$ のもの（例えば〈料理, -1 〉と〈料理, $+1$ 〉）は同時に要約に含まれないようにした。

評価情報の抽出は、パターンマッチによる評価情報抽出器によって行った。これは、文に対し係り受け解析 [9] を行い、評価表現辞書 [1] を用いて係り受け木から評価属性と評価極性の組を抽出するものである。

なお、同義関係にある評価属性がある場合には冗長な要約が生成される恐れがある。そのため、抽出した評価属性の集合をクラスタリングし、類似した評価属性は同一のものとして扱う。

3.3 接続性スコア

重要文に並びを与える際、先に挙げた [2] は文書の投稿日時を利用しているが、新聞記事を対象としており評価文書における有効性は疑わしい。[10] はコーパスを利用して文の接続確率を計算するもので、評価文書においても効果があると思われる。そのため、本稿では [10] を改良し利用する。本稿では接続性スコア $\text{Connect}(S)$ を以下のように定義する。

$$\text{Connect}(S) = \sum_{i=0}^n \text{Adjacency}(s_i, s_{i+1}) \quad (3)$$

$\text{Adjacency}(s_i, s_{i+1})$ は文 s_i の後に文 s_{i+1} が並んでいるときの接続性スコアである。すなわち文の順列 S 全体の接続性スコア $\text{Connect}(S)$ は全ての隣接する2つの文の接続性スコアの和とする。 $\text{Adjacency}(s_i, s_{i+1})$ は以下のように定義する。

$$\text{Adjacency}(s_i, s_{i+1}) = \mathbf{w}^\top \phi(s_i, s_{i+1}) \quad (4)$$

ここで、 \mathbf{w} は重みパラメタであり、 $\phi(s_i, s_{i+1})$ は文 s_i の後に文 s_{i+1} が続く並びの素性ベクトルである。すなわち、隣接する2つの文に適当な素性ベクトルを定義し、それらの重みを推定することができれば、文の順列 S の接続性スコアを計算できる。

文の並びの素性としては、[10] によって提案された隣接する文に含まれる内容語の直積集合に加え、接続詞、固有表現も含めた直積集合を用いた。これにより、例えば、飲食店の評価文書においては冒頭の文に地名を表わす固有表現が出現しやすいといった性質を取り込むことができる。

次に、 n 個の文の集合 $\mathbf{x} = \{s_1, \dots, s_n\}$ が与えられたとき、 s_0 と s_{n+1} をそれぞれ文の順列の先頭と末尾を表わす記号として、文の並び $\mathbf{y} = s_0, s_1, \dots, s_n, s_{n+1}$ の素性ベクトルを以下のように定義する。

$$\Phi(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^n \phi(s_i, s_{i+1}) \quad (5)$$

このとき、文の集合 \mathbf{x} の並び \mathbf{y} 全体の接続性スコアは $\mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y})$ となる。訓練事例が与えられた際、文の集合 \mathbf{x} の正しい並び \mathbf{y}^* の接続性スコア $\mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y}^*)$ を、誤った文の並び \mathbf{y}' の接続性スコア $\mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y}')$ より高くするよう \mathbf{w} を学習できれば、新しく文の集合が与えられた際にそれらを読みやすく並び替えることができると期待できる。そこで、平均化パーセプトロン [5] を用いて重みパラメタ \mathbf{w} を学習する。

擬似コード 1 平均化パーセプトロン

```

Training data:  $\tau = \{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1}^T$ 
1:  $\mathbf{w}^0 = 0; \mathbf{v} = 0; i = 0$ 
2: for  $n : 1 \dots N$ 
3:   for  $t : 1 \dots T$ 
4:      $\mathbf{y}' = \text{argmax}_{\mathbf{y}} \mathbf{w}^i \cdot \Phi(\mathbf{x}_t, \mathbf{y})$ 
5:     if  $(\mathbf{y}' \neq \mathbf{y}_t)$  then  $\mathbf{w}^{i+1} = \mathbf{w}^i + \Phi(\mathbf{x}_t, \mathbf{y}_t) - \Phi(\mathbf{x}_t, \mathbf{y}')$ 
6:     else  $\mathbf{w}^{i+1} = \mathbf{w}^i$ 
7:      $\mathbf{v} = \mathbf{v} + \mathbf{w}^{i+1}$ 
8:      $i = i + 1$ 
9:  $\mathbf{w}^* = \mathbf{v} / (N \times T)$ 

```

擬似コード 2 Held and Karp Algorithm

```

Sentences:  $\mathbf{S} = \{s_1, \dots, s_n\}$ 
Adjacency matrix:  $\mathbf{M} = [a_{i,j}]_{i=0 \dots n+1, j=0 \dots n+1}$ 
1:  $\mathbf{H}^0(\{s_0\}, s_0) = 0$ 
2: for  $i : 0 \dots n - 1$ 
3:   for  $j : 1 \dots n$ 
4:     foreach  $\mathbf{H}^i(\mathbf{C} \setminus \{j\}, k) \in \mathbf{b}$ 
5:        $\mathbf{H}^{i+1}(\mathbf{C}, j) = \max_{\mathbf{H}^i(\mathbf{C} \setminus \{j\}, k) \in \mathbf{b}} \mathbf{H}^i(\mathbf{C} \setminus \{j\}, k) + \mathbf{M}_{k,j}$ 
6:  $\mathbf{H}^* = \max_{\mathbf{H}^n(\mathbf{C}, k)} \mathbf{H}^n(\mathbf{C}, k) + \mathbf{M}_{k, n+1}$ 

```

擬似コード 1 に平均化パーセプトロンによるパラメタの学習アルゴリズムを示す。 T 個の訓練事例を一つずつ取り上げ (3 行目)、現在のパラメタ \mathbf{w} においてスコアが最大となる文の並び \mathbf{y}' を求める (4 行目)。もし \mathbf{y}_t と \mathbf{y}' が異なるのであれば、パラメタを更新する (5 行目)。その際、 \mathbf{v} に \mathbf{w} を加えておく (7 行目)。これを N 回繰り返す、最後に \mathbf{v} を \mathbf{w} を足した回数で割り、最終的なパラメタ \mathbf{w}^* とする。

このとき argmax 操作 (4 行目) は、文を都市とみなすと、 s_0 を始点、 s_{n+1} を終点として s_1 から s_n までの全ての都市を通過する、接続性スコアが最大の経路を求める巡回セールスマン問題となる。そのため、接続性スコアが最大となる経路の探索は容易でない。ここでは、動的計画法の一種である Held and Karp Algorithm [7] とビームサーチを用いて近似解を探索する。

擬似コード 2 に Held and Karp Algorithm による探索の手順を示す。並び替えの対象とする文の集合 \mathbf{S} とそれらの文の間の接続性スコア \mathbf{M} が与えられているとする。 $\mathbf{H}^i(\mathbf{C}, j)$ は $i+1$ 個の文 \mathbf{C} を網羅し、最後に追加した文が $j \in \mathbf{C}$ である仮説の接続性スコアを表わす。1 行目に示す $\mathbf{H}^0(\{s_0\}, s_0)$ は、既に文 s_0 を網羅しており、かつ最後に追加した文が s_0 である仮説を示す。この仮説の接続性スコアを 0 に初期化し、以降、逐次的にこの仮説に新しい文を追加していく。基本的な戦略は、既に網羅した文の集合 \mathbf{C} が同一であり、かつ最後に追加した文 j が同一である仮説は、接続性スコアが高いもののみを残すというものである (動的計画法)。このとき、全探索は難しいことから、上位 b 件の仮説 \mathbf{b} のみを展開していく。

3.4 探索

接続性スコアの学習と同様に、式 1 の argmax 操作も容易ではない。そのため、動的計画法とビームサーチを用いて近似解

を求める。基本的なアルゴリズムは擬似コード 2 で示したものと同様であるが、Connect(S)と同時に Content(S)を計算しなければならないこと、仮説のサイズが K 以内に制限されていることを考慮する必要がある。そのため、仮説に新しい文を追加した際には Content(S)を再計算する。サイズに関しては、仮説に新しい文を追加する際、制限サイズを超える文は追加を行わず、いかなる文を追加しても制限サイズを超える仮説は別途保存し以降新しく文を追加しない。全ての探索が終わった後、保存された仮説のうちスコアが最大のものを出力する。

4 評価と考察

4.1 実験の設定

本節では、提案手法を内容性と可読性の観点から評価する。評価文書のドメインは飲食店とし、飲食店の評価文書が多く投稿されているウェブサイトから 100 店舗に紐づく 2940 文書を収集した。1 店舗に紐付いている文書の平均サイズは 5343 バイトであるため、それぞれの店舗に対して 300 バイトの要約を生成する場合、要約率は約 6% となる。

4.2 内容性の評価

内容性の評価には ROUGE[12] を用いる。本稿では ROUGE-2, ROUGE-SU4, ROUGE-SU9 を利用した。評価に際しては評価文書を 4 名の作業者に要約させ、各店舗に 4 つの参照要約を付与した。システム要約の ROUGE スコアは 4 つの参照要約との ROUGE スコアの平均を取ったものである。

ROUGE に用いる N グラムは、ROUGE-2, ROUGE-SU4, ROUGE-SU9 いずれも全ての単語を用いて N グラムを計算した。また、 N グラムを計算する際には、参照要約、システム要約共に文に分割せず、それぞれを 1 つの長い文と見なして N グラムを計算した。これによって、ROUGE-SU4 や ROUGE-SU9 といった 2 つの単語の長距離の並びを考慮する尺度を用いて文間の単語の並びを考慮することができるようになる。

本稿では、評価情報、組み合わせ最適化、接続性スコアの効果をそれぞれ検証するため、以下の 4 種類の手法を比較する。

表層 (Baseline) 評価情報を利用せず、文のスコアとして内容語のスコアの和を用いる。内容語のスコアは、要約の対象とする文書集合中の単語頻度とした。重要文を選ぶ際には MMR[3] を用いて冗長性を排除しつつ貪欲に文を選択した。MMR の類似度にはコサインを用い、新しく選ぶ文の重要度と、既に選んだ文との類似度のトレードオフを制御するパラメタは 0.5 とした。

表層+組み合わせ最適化 (Method1) 評価情報を利用せず、Baseline と同様に内容語のスコアを用いる。重要文を選ぶ際には MMR による貪欲法ではなく組み合わせ最適化を用い、スタック・デコーダ [13] を用いて探索を行った。冗長性の削減のため、各内容語は 1 度しか要約のスコアに加算されないようにした。スタックのサイズは 10 とした。

評価情報+組み合わせ最適化 (Method2) 評価情報を利用する。接続性スコアは利用せず、式 1 において $\lambda = 0$ としたものに相当する。探索には Method1 同様スタック・デコーダを利用した。なお、Baseline, Method1, Method2 の文の並びについては、抽出元の文書の投稿日時が新しく、かつその文書において前に現れる文から並べるようにした。

表 1 手法別の ROUGE スコア

	ROUGE-2	ROUGE-SU4	ROUGE-SU9
Baseline	0.089	0.068	0.062
Method1	0.157	0.096	0.089
Method2	0.172	0.107	0.098
Method3	0.180	0.110	0.101
Human	0.258	0.143	0.131

表 2 要約に含まれる評価情報の異なり数

	評価情報の異なり数
Baseline	1.76
Method1	4.32
Method2	10.41
Method3	10.18
Human	4.75

評価情報+組み合わせ最適化+接続性スコア (Method3) 提案手法。式 1 に従い、内容性スコアと接続性スコアの和が高い文の順列を探索する。接続性スコアの重みパラメタ w は、参照要約からではなく、要約の対象として収集した文書から 5 分割交差検定にて学習したが、前述した 100 店舗とは別の 150 店舗に紐づく評価文書も訓練セットに加えた。 λ は開発セットで調整し、6000 とし、ビーム幅は 100 とした。

参照要約 (Human) 人間の要約のパフォーマンスを調べるため、参照要約間の ROUGE スコアを計算する。具体的には、4 つの参照要約のうち、いずれか 1 つと残りの 3 つに対して ROUGE スコアを計算する。これを 4 つの参照要約それぞれに行い、平均を取る。

結果を表 1 に示す。全ての尺度において、Baseline, Method1, Method2, Method3 の順にスコアが向上するが、Human には及ばない。ウィルコクソンの符号順位検定によれば、Baseline に対し他の全ての手法は有意に ROUGE スコアが向上している。

接続性スコアの ROUGE への効果を議論する。Method2 と Method3 を比較すると、全ての尺度において ROUGE スコアが向上している。ROUGE-SU4 や ROUGE-SU9 といった尺度は長距離の単語の並びを考慮するため参照要約に近い文の並びを持つシステム要約に高い ROUGE スコアを与えられると考えられる。そのため Method3 がこれらの尺度においてスコアが高いことは直観的である。一方、Method3 は ROUGE-2 スコアも他の手法より高い。これは、接続性を考慮すると、内容性のみよりも適切に文を選択できることを示唆している。

表 2 にシステム要約に含まれる評価情報の異なり数を示す。これは各手法が生成した要約に対し評価情報抽出を行い、各要約が含む評価情報の異なり数の平均を取ったものである。

評価情報の異なり数については、参照要約に対し、システム要約は 2 倍以上の評価情報を含んでいる。換言すると、参照要約はシステム要約の半分しか評価情報を含んでいないが、その分別の情報が要約に含まれていると考えられる。参照要約には、店舗の概要を伝える文や評価情報の根拠となる表現、利用すべ

表3 可読性評価の得点

	可読性得点
Method2	3.45
Method2+	3.54
Method3	3.74

きシチュエーションなど、評価情報の抽出のみでは捉えられない情報が含まれている。これらの情報も要約に含めていくことが ROUGE スコアの向上に向けた今後の課題となる。

4.3 可読性の評価

可読性は被験者実験にて評価した。前述の 100 店舗に紐づく評価文書に対し 3 種類の方法で要約を生成し、計 300 のシステム要約を用意した。10 名の被験者を用意し、1 人の被験者に 30 のシステム要約を評価させた。評価方法は提示されたシステム要約に対し 5 段階の評価 (1 点から 5 点) を与えるものである。被験者には、実験前に要約の品質の評価基準を提示し、評価基準に従って要約を評価するように指示した。また、被験者には要約を生成した手法は通知されない。

評価の対象として、重要文を選んだ後に抽出元の文書の投稿日時と文書内での位置に基づいて文を並び替えるもの (**Method2**)、重要文を選んだ後に前述の接続性スコアを用いて文を並び替えるもの (**Method2+**)、直接文書集合から文の順列を探索するもの (**Method3**) を用意した。

評価の結果を表 3 に示す。Method2, Method2+, Method3 の順に可読性が向上した。ウィルコクソンの符号順位検定にて検定を行ったところ、Method2 と Method2+ に有意差は見られず、Method2 と Method3 は $p < 0.10$ で有意であった。このことから、まず文を選んだ後に文を並び替えるという方法では必ずしも可読性は向上せず、最初から文の並びを考慮した場合には可読性が向上すると考えられる。

可読性が向上した要因として、要約に含まれる文数の変化が考えられる。Method2 および Method2+ が平均 7.23 文で構成されるのに対し、Method3 は平均 6.52 文である。この差は $p < 0.01$ で有意である。すなわち Method2 および Method2+ は短い文を選択しやすく、結果としてリストのような要約が生成されることが多く読みづらいが、Method3 ではより長い文が選択されやすく、読みやすさが向上したと考えられる。

また、全てのシステム要約において、同一の文書の中で隣接していた文がそのままの形で並んでいる回数は、Method2 が 0 回であったのに対し Method3 は 12 回であった。このことから、接続性を考慮し文を選択することによって、文よりいくらか広い単位で元の文書から情報が抽出されることが増え、結果として要約の可読性が向上したということも考えられる。

5 まとめ

本稿では文の内容及び接続性を共に考慮し、複数の評価文書を要約する手法を提案した。評価文書を要約の対象とすることから内容性に関しては評価情報を用いて計算し、接続性については同一ドメインの文書から学習しておく。スコアの高い文の順列は動的計画法とビームサーチを用いて探索する。これによ

り、内容性、可読性いずれの観点からも高い品質を持つ要約が生成されることを示した。

今後の課題は 2 つある。1 つは前述したように要約に含めるべき評価情報以外の情報の特定である。もう 1 つはより大域的な接続性のモデル化である。今回用いたモデルは単に前後の文を見るに過ぎない局所的なモデルであるが、より大域的なモデルがより良い要約には重要と考えられる。

謝辞

本研究の推進にあたり NTT コミュニケーション科学基礎研究所 平尾努氏より多大なご助言をいただきました。深く御礼申し上げます。

参考文献

- [1] 浅野久子, 平野徹, 小林のぞみ, 松尾義博. Web 上の口コミを分析する評判情報インデクシング技術. NTT 技術ジャーナル, Vol.20, No.6, pp. 12-15, 2008.
- [2] Regina Barzilay, Noemie Elhadad and Kathleen McKeown. Inferring Strategies for Sentence Ordering in Multidocument Summarization. *Journal of Artificial Intelligence Research (JAIR)*, Vol.17, pp. 35-55, 2002.
- [3] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, pp. 335-356, 1998.
- [4] Giuseppe Carenini, Raymond Ng and Adam Pauls. Multi-Document Summarization of Evaluative Text. In *Proceedings of the 11th European Chapter of the Association for Computational Linguistics (EACL)*, pp. 305-312, 2006.
- [5] Michael Collins. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the 2002 Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pp. 1-8, 2002.
- [6] Elena Filatova and Vasileios Hatzivassiloglou. A Formal Model for Information Selection in Multi-Sentence Text Extraction. In *Proceeding of the 20th International Conference on Computational Linguistics (COLING)*, pp. 397-403, 2004.
- [7] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics (SIAM)*, Vol.10, No.1, pp. 196-210, 1962.
- [8] Mingqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. In *Proceeding of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 168-177, 2004.
- [9] 今村賢治. 系列ラベリングによる準話し言葉の日本語係り受け解析. 言語処理学会第 13 回年次大会発表論文集, pp. 548-551, 2006.
- [10] Mirella Lapata. Probabilistic Text Structuring: Experiments with Sentence Ordering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 545-552, 2003.
- [11] Kevin Lerman, Sasha Blair-Goldensohn and Ryan McDonald. Sentiment Summarization: Evaluating and Learning User Preferences. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 514-522, 2009.
- [12] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*, pp. 74-81, 2004.
- [13] Wen-tau Yih, Joshua Goodman, Lucy Vanderwende and Hisami Suzuki. Multi-Document Summarization by Maximizing Informative Content-Words. In *Proceeding of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1776-1782, 2007.