スレッド構造と語彙的連鎖を用いたオンラインディスカッションからの重要文・トピックの抽出

羽鳥 潤
東京大学大学院 情報理工学系研究科
東京都文京区本郷 7-3-1
hatori{at}is.s.u-tokyo.ac.jp

村上明子
IBM 東京基礎研究所
神奈川県大和市下鶴間 1623-14
akikom{at}jp.ibm.com

1 序論

近年、オンラインディスカッションの普及に伴い、膨大な量のディスカッションがウェブ上で行われるようになった。それに伴い、ディスカッションの内容を分かりやすく概観できるシステムへの需要は日に日に増している。このようなシステムを構築する際には、新聞記事などの一般のテキストに適用される複数文書要約手法を利用するアプローチも考えられるが、この種のコーパスには一般のテキストとは異なった特徴があり、そのままでは通常の要約手法が適切に働かないおそれがある。

そのような特徴の一つとして、まず、一つの議論のまとまり（スレッド）において多数のトピックが議論されている事が挙げられる。ディスカッションの内容を概観するためには、このようなトピックのうち主要なものが網羅されている事が望ましい。しかし、今日用いられている典型的な要約手法では文の情報量の大きさが主たる基準として用いられており、全ての主要なトピックが網羅される事は保証されていない。また、そもそも、このように多数のトピックが並立するコーパスにおいては、固定長の要約文を生成するよりは、トピックの構造が明確に閲覧できた方が良い可能性がある。また、第二の特徴として、オンラインディスカッション内の多くの発言は非常に短い事が挙げられる。このため、それぞれ発言に至った経緯や話題の流れを見ることなしには、発言者の主張を明確に捉える事が難しい。一方で、各発言者は短い発言の中で自らの主張を明確にするために特徴的で分かりやすい表現を使う傾向があり、そのような表現を適切に扱う事も必要であると考えられる。

これらの問題点を踏まえて、まず、本研究ではスレッド全体の要約文を生成するアプローチを取らず、各発言の重要文とスレッドのトピックを抽出するという 2 つのタスクに焦点を置き、スレッド全体をトピックの観点から概観する事を目指した。我々は、IBM 社内のディスカッションである InnovationJam[*1]のコーパスに人手で重要文とトピックの情報を付与し、これを実験に用いた。図 1 に InnovationJam（以下 Jam）のスレッドの一例が示されている。ここで、各発言は返答構造によって接続されており、また、各発言の中心的な文（重要文）には下線が引かれている。

我々は、Wan (2008) による既存の重要文抽出手法に、この種のコーパスに有効であると考えられる 3 つの要素を加えてモデルを構築した。まず、図 1 中に斜体で示されているように、重要文には “I think”・“should” 等の発言者の意見や提案に特徴的な表現が頻出している事に注目した。これらは、言語学における**手がかり語**(Edmundson, 1969) の一種であると考えられる。我々はこのコーパスに有効な手がかり語を調査し、重要文抽出のモデルに用いた。

[*1] http://www.ibm.com/ibm/jam/

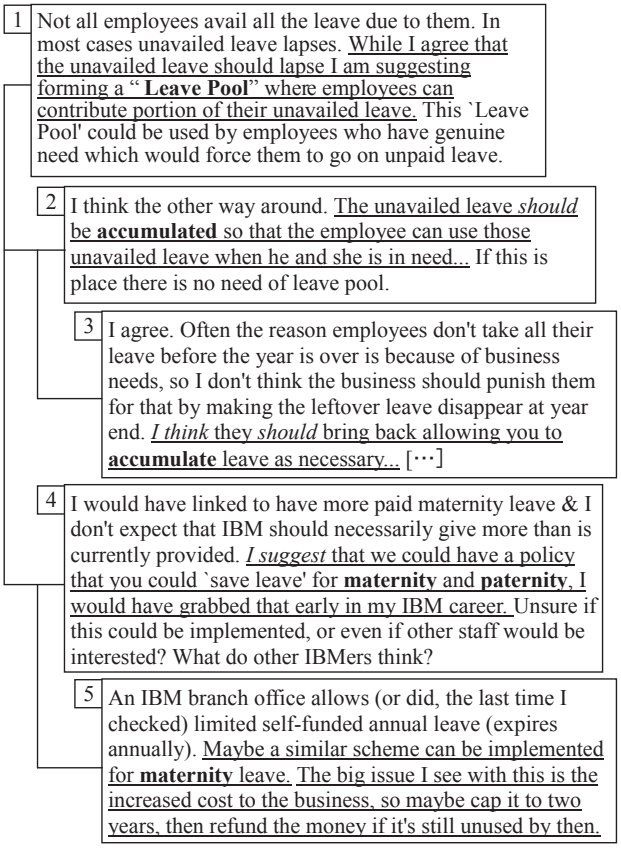

図 1 Jam のスレッドの例

また、次に、親子関係にある発言は話題的な繋がりが強いという傾向、及びその繋がりは話題に関連した語句の連鎖として出現する傾向があることに着目した。図 1 の中で、発言 1 で提起された「休暇のプール」に関するテーマは、発言 2・3 における「休暇の積み立て」を主張する対案と、発言 4・5 における「育児休暇」に言及した議論に発展している。ここでは、親子発言間の話題に強い繋がりがある事と、各話題の流れが、それぞれ “leave [pool]”・“accumulate(d)’・“maternity, paternity” という関連語の連鎖として出現している様子がはっきりと見て取れる。このように、我々は、返答関係にある発言間に強い依存関係を考えるとともに、関連語の出現として現れる**語彙的連鎖**(Morris and Hirst, 1991) がディスカッションのトピックを表していると仮定し、これを重要文・トピック抽出の両タスクで用いた。

ここで、本稿で用いられるディスカッションの用語を整理しておく。

フォーラム: ある大きなテーマについて、参加者が様々な問題・論点について議論を交わす場。

スレッド: フォーラムの中で、ある特定の問題・論点について議論している一連の発言。スレッド内では、各発言は返答関係に基づく木構造を持つ。
発言: 参加者によって投稿された一つのメッセージ。

以下の章では、まず、第 2 節で主要な関連研究を紹介した後、第 3 節でタスク設定と提案手法を詳説する。その後、第 4 節で実験の概要と結果、第 5 節で結論を示す。最後に、今後の課題や研究の方向性について、第 6 節で簡単に議論する。

2 関連研究

この項では、我々の研究に関連したいくつかの研究を紹介する。まず最初に、対象コーパスの性質が近い、ブログ・メーリングリスト・ディスカッションなどのウェブテキストに対するアプローチを紹介する。次に、この種のコーパスには応用されていないが、今回我々が用いる語彙的連鎖に関連した研究を紹介する。最後に、我々の手法の基盤となる、複数文書要約に用いられるグラフベースの手法を紹介する。

まず、ウェブテキストを対象とした要約・概観システムを構築する際には、膨大な量の情報がリアルタイムで増えていく点と、コーパスを構成するドキュメント間の返答関係や話題の流れをいかに扱うかが焦点となる。一つのアプローチは、各ドキュメントの重要度をスレッド構造を元に評価する事によって、読むべきドキュメントの絞り込みを可能にした、Klaas (2005); Murakami et al. (2007) の研究である。また、別のアプローチとして、スレッド全体の要約文を生成する方法も研究されており、例えば、メーリングリストに対する Lam et al. (2002) の研究、ディスカッションに対する Carenini et al. (2007) の研究がある。Lam et al. (2002) は、各発言のコンテキストとして、そのスレッド構造上の先祖となっている発言集合を利用して、要約文の生成を行った。また、Carenini et al. (2007) は、スレッド中での同一語の連続的な出現（clue word）をディスカッションの要約に利用することで、スレッドの構造を間接的に用いている。しかし、これらの研究ではドキュメント間の返答構造は間接的に用いられており、個々の依存関係が陽に考慮されてはいない。

語彙的連鎖は、Carenini et al. (2007) が用いた clue word を、意味的に関連した語の出現にまで一般化した概念である。語彙的連鎖は、我々の知る限りこのようなウェブテキストの要約に応用されていないが、単一ドキュメントの要約においては、Barzilay and Elhadad (1997); Brunn et al. (2001) 等の利用例がある。語彙的連鎖の構築には、Silber and McCoy (2002) による効率的な手法が存在する。彼らの手法は、WordNet(Fellbaum, 1998) をベースとした簡単な語義曖昧性解消を行い、単語数に対して線形時間で語彙的連鎖を構築する事を可能にしている。また、Ercan and Cicekli (2007) は、強い語彙的連鎖はテキストの一つの主要な話題を表しているという仮定の下に、単一ドキュメントからのキーワード抽出を行っている。しかし、語彙的連鎖をウェブテキストの持つスレッド構造と組み合わせて利用した研究は、我々の把握している限り存在しておらず、この点は本研究の大きな貢献となっている。

最後に、一般的な複数文書要約モデルの一つとして、Mihalcea and Tarau (2005) によるグラフベースの手法を紹介する。彼らの手法は、パラメータのチューニングの必要がほとんどない非常にシンプルな枠組みにも関わらず、対象言語によらない一般性と最高精度の性能を持っており、今回我々のモデルのベースとして使用する事にした。このモデルは、各文が頂点に、2 文によって共有される単語が枝に対応したグラフ（木）を構築した後、各頂点の PageRank(Page et al., 1999) を計算することで、スコアの高い文を各文書の重要文として出力する。また、この手法には Wan (2008) による拡張が存在し、彼はこの枠組みの上に各文書の重要度と文・ドキュメント間の類似度等を考慮に入れることで、複数文書の要約に適したモデルを構築した。

3 提案手法

3.1 タスク設定

本稿では、**(1) 各発言の重要文・(2) 各スレッドのトピック**を抽出するという 2 つのタスクに取り組み、データセットの作成とモデルの構築・評価を行った。

重要文は、各発言の最も中心的な文章で、ディスカッションでは発言者の提案や主張がその発言の重要文となる事が多い。図 1 では各発言の重要文が下線で示されているが、発言ごとにその重要文を抽出する事がこのタスクの目的となる。データセット作成の際には、文の強勢の度合いに加えて情報量の大きさも考慮し、例えば、単純な同意の表現（“I agree with your idea.” など）より、具体的な主張や意見（“We should ...” など）を優先するなどした。

トピックは、スレッドの中で議論されている一つの論点・話題である。一つのトピックはキーワードの集合として定義され、例えば図 1 の例では、“leave”・“accumulate”・“maternity, paternity” がそれぞれトピックの定義語となる。トピック抽出タスクでは、各スレッドの主要なトピックを、このようなキーワードの集合として抽出する事を目的とする。

以下の項では、重要文・トピック抽出に用いた我々の提案手法を紹介する。

3.2 重要文抽出モデル

重要文抽出モデルを構築するため、はじめに、前節で紹介した Mihalcea and Tarau (2005) と Wan (2008) によるグラフベースのモデルを構築した。この際、予備実験の結果寄与の認められなかった文・文書間の類似度は無視することにし、最終的に以下の式に基づいて PageRank $R(s)$ を計算した（d=0.5 とした）。

$$R(s) = (1-d) + d \sum_{s' \in \mathcal{S}} \frac{f(s,s')R(s')}{\sum_{s'' \in \mathcal{S}} f(s,s'')} \quad (1)$$

$$f(s,s') = \mathrm{Sim}(s,s') \frac{\mathrm{Imp}(s) + \mathrm{Imp}(s')}{2} \quad (2)$$

$$\mathrm{Imp}(s) = \mathrm{Sim}(s, doc(s)) \quad (3)$$

このモデルの上に、我々は手がかり語・スレッド構造・語彙的連鎖の 3 要素を加えて、その寄与を調べた。

3.2.1 手がかり語

手がかり語は、重要文に特徴的な表現である**ボーナス語**と、逆に非重要文に特徴的な**スティグマ語**に分けられる。例えば、“important”・“should”・“I propose” などの語句は筆者の意見が書かれた文に特徴的でボーナス語の一例であると考えられるが、逆に “for instance” などの表現は具体例を挙げて主張を補足する際に用いられる表現で、スティグマ語の一例であると考えられる。開発セットから、我々は 31 の手がかり語を抽出し、それぞれに人手で重みを付与した。実験で用いた手がかり語の一覧を、表 1 に示す。

ボーナス語	should, would, could, important, significant, real, now, proposal, idea, challenge, conclusion, suggest, propose, believe, need, thus, therefore, so, for this reason, I, my, it's, so there, problem be, point be, be to, one thought be
スティグマ語	example, for example, for instance, agree

表 1　実験に用いた手がかり語の一覧

手がかり語を考慮する場合、枝の重みは、

$$\mathrm{Imp}(s) = \mathrm{Sim}(s, doc(s)) \prod_{c \in \mathcal{W}(s)} \mathrm{CueScore}(c) \quad (4)$$

となる。ここで、$\mathcal{W}(s)$ は文 s 内の手がかり語の集合である。

3.2.2　スレッド構造

返答構造にある発言間の依存関係は、文間の枝に対する重みとして表現される。この時式 (2) は、

$$f(s, s') = \mathrm{Sim}(s, s')\mathrm{Rel}(s, s')\frac{\mathrm{Imp}(s) + \mathrm{Imp}(s')}{2} \quad (5)$$

となる。ここで、$Rel(s, s')$ は、文 s・s' が同一発言内に含まれる時 2.0、親子関係にある発言に含まれる時 1.5、それ以外の時 1.0 とした。また、スレッド構造は後に記述するように、語彙的連鎖から局所的な部分鎖を見つけ出す際にも用いられる。我々のモデルは、Lam et al. (2002) と比べると、各発言間の依存関係が明確に表現されている点で、より精細なモデリングを行っていると考えられる。

3.2.3　語彙的連鎖

我々は、Ercan and Cicekli (2007) に倣い、一つの語彙的連鎖がスレッドの一つのトピックを表現していると仮定した。語彙的連鎖の構築には、第 2 節で紹介した Silber and McCoy (2002) による手法を用いたが、彼らの用いた同義語・上位語・下位語・兄弟語のリンク構造に加えて、新たに holonym・meronym・名詞化の 3 種類のリンクを加えている。

語彙的連鎖の寄与を加えると、式 (5) は

$$f(s, s') = \mathrm{Sim}(s, s')\mathrm{Rel}(s, s')\frac{\mathrm{Imp}(s) + \mathrm{Imp}(s')}{2} + \lambda \sum_{c \in \mathcal{LC}(s, s')} \mathrm{Score}(c) \quad (6)$$

となる。ここで、$\mathcal{LC}(s, s')$ は文 s と文 s' の両方に跨った語彙的連鎖の集合である。λ の値としては 0.5 を用いた。語彙的連鎖のスコア $\mathrm{Score}(c)$ の計算については、次項で詳しく説明する。

3.3　トピック抽出モデル

トピック抽出タスクにおいては、重要文抽出モデルにより構築された語彙的連鎖が、スコア $\mathrm{Score}(c)$ の高い連鎖から順に、スレッドの主要なトピックとして出力される。我々は、スレッド構造に基づいて、より適切な連鎖のスコアリングを行う指標を提案する。

まず、スレッド構造の中で局所的に現れる連鎖である**部分鎖**の概念を導入し、語彙的連鎖を部分鎖の集合として定義した。ある語彙的連鎖の構成語が親子関係にある発言の両方に出現したら、それらを**直接枝**(図 2 の e_1, e_3, e_4, e_5) にて接続する。また、親子間での連続性は

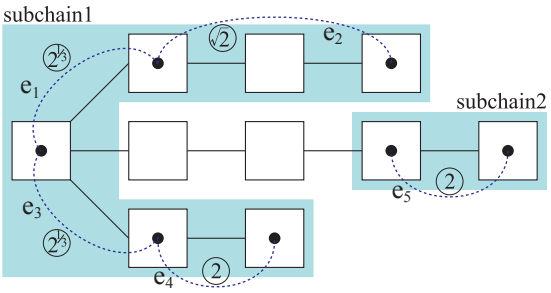

図 2　語彙的連鎖のスコア付けの例。黒丸は正方形で表される発言の中に対象の語彙的連鎖の構成語が出現している事を示す。

ないが、祖父母・孫関係にある発言の両方に構成語が出現したら、それも局所的な連鎖の一部であると仮定し、それらを**間接枝**(図 2 の e_2) にて接続する。このように、語彙的連鎖の構成語が出現する発言可能な限り接続して行き、最終的に語彙的連鎖を構成する部分鎖の集合を得る。この結果、図 2 は最終的に 2 つの部分鎖からなる事が分かる。

語彙的連鎖 c に対するスコア $\mathrm{Score}(c)$ は、

$$\mathrm{Score}(c) = \mathrm{Strength}(c) + \mathrm{Locality}(c) \quad (7)$$

$$\mathrm{Locality}(c) = \ln \sum_{c' \in \mathcal{SC}(c)} \prod_{e \in \mathcal{E}(c')} \mathrm{EScore}(e) \quad (8)$$

$$\mathrm{EScore}(e) = \begin{cases} 2^{\frac{1}{n(e)}} & e \text{ が直接枝の時} \\ 2^{\frac{1}{2n(e)}} & e \text{ が間接枝の時} \end{cases} \quad (9)$$

として計算される。$\mathcal{SC}(c)$ は語彙的連鎖 c に対する部分鎖の集合、$\mathcal{E}(c')$ は部分鎖 c' に属する直接・間接枝の集合、$n(e)$ は枝 e の始点 (e_{start}) に対応する語を含む文書の兄弟の数である。ここで、語彙的連鎖の局所性の強さを表す locality という指標を導入し、スレッドの中で局所的に集中して出現している語彙的連鎖は強いトピックであるという考えを導入している。また、語彙的連鎖の強さ $\mathrm{Strength}(c)$ の計算は、Silber and McCoy (2002) の手法に準じている。

図 2 は、locality の算出法を具体的に示している。この例に対しては、locality は $\ln\left[(2^{\frac{1}{3}} \cdot \sqrt{2}) \cdot (2^{\frac{1}{3}} \cdot 2) + 2\right]$ として計算される。

4　実験

4.1　実験条件

InnovationJam 2008 のデータから 10 スレッドに重要文・トピックの情報を付与し、このうち 5 スレッドを開発セットとして、5 スレッドをテストセットとして実験に用いた。1 発言あたりの平均重要文数は 1.54、1 スレッドあたりの平均トピック数は 4.10 だった。また、2 人の注釈者間の一致率を 3 スレッドに対して測定したところ、概ね（最）重要文に対して 70% 程度、トピックに対して 60% 程度の一致率が得られた。

重要文抽出タスクの評価のために、Wan (2008) の手法を再実装してベースラインモデルとした。各モデルは、各発言の中で最もスコアの高い文を最重要文として出力し、評価はこれが正解の重要文と一致するかどうかによった。

トピック抽出タスクに対しては、標準的な **TF-IDF** と、グラフの**枝スコア**に基づく 2 つの指標をベースラインとして用いた。ここで、**枝スコア**とは、重要文抽出の際に

	精度（開発）	精度（テスト）
ベースライン	56.9% (91/160)	57.1% (105/184)
+ 手がかり語	60.6% (97/160)	62.0% (114/184)
+ 語彙的連鎖	64.4% (103/160)	64.1% (118/184)

表 2　重要文抽出の結果

	ミクロ平均	マクロ平均
TF-IDF	19.51% (8/41)	23.08%
枝スコア	24.39% (10/41)	25.92%
語彙的連鎖	36.59% (15/41)	34.17%

表 3　トピック抽出の結果（再現率）

用いたグラフに基づくもので、各頂点の PageRank $R(s)$ を計算した後、以下のようにして求められる。

$$S(w) = \ln \frac{\#doc}{\mathrm{DF}(w)} \sum_{e \in \mathcal{E}(w)} R(e_{\mathrm{start}}) R(e_{\mathrm{end}}) \qquad (10)$$

ここで、$\mathcal{E}(w)$ は語 w に対応したエッジの集合、$\mathrm{DF}(w)$ は語 w の文書頻度である。各スレッドにおける正解トピックの数はモデルに入力として与えられ、モデルはスコアの高い順に正解トピック数と等しい数のトピックを出力する。評価は、出力トピック集合が正解トピック集合をどれだけカバーしているかに基づく再現率を基準として行った。

4.2　結果

表 2 は、重要文抽出の結果を示している。まず、人手で設定した少数の手がかり語を利用したモデルは、ベースラインモデルを大幅に上回る性能を見せた。また、さらに語彙的連鎖の寄与を加えることで、更なる精度向上が得られている。しかし、ここには結果は示されていないが、我々の予想に反してスレッド構造を単独で用いることによる精度向上は確認されなかった。

表 3 はトピック抽出の結果を示している。語彙的連鎖に基づいたモデルは 2 つのベースラインモデルを大幅に上回る精度を記録し、語彙的連鎖をこのタスクに用いることの有用性を示していると言える。

しかし、これらの実験はコーパスのサイズを考えると予備実験の域を出ず、これらの結果を検証するためには、さらに大きなコーパスでの実験が必要であると考えられる。

5　結論

本稿では、オンラインディスカッションにおける議論の内容を重要文とトピックから概観する手法を提案した。特に、このようなコーパスの特徴を的確に捉えるため、手がかり語・スレッド構造・語彙的連鎖の 3 要素をモデルの中核に据えた。

実験の結果、語彙的連鎖と手がかり語がこのタスクに有用であることが示された。スレッド構造を単独で用いることによる精度向上は確認できなかったが、この構造は、語彙的連鎖の構築時に重要な役割を果たした。

これらの手法を他のデータセットやタスクに適用する事は興味深い。今回の実験は、IBM 社内の InnovationJam のデータに対して行われたものだが、スレッド構造を持つような他のコーパスにもそのまま適用する事ができると考えられる。

6　今後の課題

現在の実験設定における最大の問題点は、実験に使用したコーパスが非常に小さいことである。今回の実験で得られた結果をより明確に示すためには、更に大きなコーパスでの実験が必要であろう。我々は、今後、他のオンラインディスカッションやメーリングリストのコーパスを実験に使用することを計画している。

また、語彙的連鎖を構築する際に使用した関連語同士のリンク構造は、現在のところ WordNet 内で定義されているものに限定されており、専門性の高い用語の多く出現する Jam のデータにおいて、適切な関係を認識できなかった例が多く見つかった。そのため、今後、生コーパスから上位・下位語などのリンク構造を自動抽出する手法を用いて、より被覆率の高い語彙的連鎖を構築できるようにしたい。

最後に、今回の実験では、人手で 30 個程度の手がかり語を指定することで重要文抽出の大幅な精度向上を確認することが出来たが、これらを教師あり学習、またはブートストラップ法により自動学習することは興味深い。これは、精度向上のためだけでなく、有効に働く手がかり語が大きく異なる可能性のある他のドメインのテキストに手法を適用する上で、非常に重要な要素である。

参考文献

Barzilay, Regina and Michael Elhadad. 1997. Using lexical chains for text summarization. In *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

Brunn, Meru, Yllias Chali, and Christopher Pinchak. 2001. Text summarization using lexical chains. In *Document Understanding Conference (DUC)*, pages 135–140.

Carenini, Giuseppe, Raymond T. Ng, and Xiaodong Zhou. 2007. Summarizing email conversations with clue words. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 91–100.

Edmundson, H.P. 1969. New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285.

Ercan, Gonenc and Ilyas Cicekli. 2007. Using lexical chains for keyword extraction. *Inf. Process. Manage.*, 43(6):1705–1714.

Fellbaum, C. 1998. Wordnet: An electronic lexical database.

Klaas, Mike. 2005. Toward indicative discussion fora summarization. In *UBC CS TR-2005-04*.

Lam, Derek, Steven L. Rohall, Chris Schmandt, and Mia K. Stern. 2002. Exploiting e-mail structure to improve summarization.

Mihalcea, Rada and Paul Tarau. 2005. A language independent algorithm for single and multiple document summarization. In *In Proceedings of IJCNLP-2005*.

Morris, J. and G. Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–43.

Murakami, Akiko, Tetsuya Nasukawa, and Hiroshi Nakagawa. 2007. オンラインディスカッションにおける有益発言の抽出. In *Proceedings of the 14th meeting of the association for natural language processing*, pages 352–355.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web.

Silber, H. Grogory and Kathleen F. McCoy. 2002. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4):487–496.

Wan, Xiaojun. 2008. An exploration of document impact on graph-based multi-document summarization. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 755–762, Morristown, NJ, USA. Association for Computational Linguistics.