

PTOS：機械学習を用いた論文要旨スライド自動生成システムの提案と開発

谷口 航平^{1,a)} 濱川 礼²

概要：本論文では、機械学習を用いて論文の要旨を1枚のスライドとして自動生成するシステム PTOS について述べる。研究活動において、自身の研究テーマに関連する論文を網羅的に調査・勉強する論文サーベイは非常に重要な過程の1つである。それは自身の研究の新規性、優位性の検証や研究テーマのヒントに繋がるからである。現在の論文サーベイの手法は Google Scholar や ACM Digital Library といった論文のデータベースから研究テーマに関連するキーワードで検索し、論文を1つ1つ読み、内容をまとめるのが主である。しかしながら、キーワードの選び方や検索の仕方によっては自身の研究とは関連のない論文がヒットすることが多く、また、論文の概要を読んだ時点では関連していそうだと感じた論文でも最後まで読んだ結果関連していない論文であることがある。こういった状況で膨大な量を論文サーベイすると現状のサーベイは非効率的であると考えられる。そこで、より短時間でより多くの論文の大まかな流れを把握することができればサーベイの効率が向上すると考えた。我々は論文の本文のテキストデータを入力することで、要旨を1枚のスライドとして自動生成するシステム PTOS を開発した。PTOS は要旨として必要となる文章を5種類のカテゴリと定義し、機械学習によって論文のテキストデータに対して5クラス分類を行い、その後1枚スライドを自動生成する。また、PTOS の有効性についての評価実験の結果についても述べる。

1. はじめに

研究活動において、論文サーベイは研究動向の調査や自身の研究の新規性の確認をするためにきわめて重要な過程の1つである。現在の論文サーベイ手法は、自身の研究の関連テーマのキーワードを Google Scholar[1] や ACM Digital Library[2] といった論文のデータベースで検索し、結果として返ってきた論文を1つ1つ読んでいくといった手法である。例えば、研究分野の動向を把握する場合は、曖昧なキーワードを指定して膨大な検索結果を1つ1つ読んでいく。他にも、Github や Qiita で特定の分野の論文を集めたサイト [3][4] が存在し、そのサイトから目当ての論文を探すといった手法もある。

しかしながら、研究活動を目的とした論文執筆の経験がない研究者 (以下、研究初心者) は、多くの場合、サーベイや研究に関する知識や経験がない場合が多い。そのため、サーベイすべき研究分野や、サーベイから得られる情報の適切な整理の方法などがわからず、自身の研究分野についての理解を深めることが難しくなる。また、UIST2018 で

は 80 件、さらに CHI conference では今年だけで 666 件の論文を発表されているように論文は毎年膨大な数の増加を示しており、研究初心者だけでなく、サーベイに慣れた研究者でも論文を読むのに膨大な時間が必要である。論文を読む際に「アブストラクト」「第1章」「結論」だけを読むスタイルであれば効率的に読むことはできるが、その場合「技術の核」や「評価手法」といった重要な箇所を見逃してしまう。よって、現状のサーベイの手法では非効率であると考ええる。

そこで我々は1つ1つの論文のおおまかな内容を把握し、後で気になる論文のみをじっくり読むスタイルであればサーベイの効率が低いのではないかと考えた。具体的には、研究初心者には自身で検索した結果全てについて論文をじっくり読む必要もなく一目でどういった論文が分かるように、研究初心者ではない研究者には膨大な量の論文の大まかな内容を短時間で把握できれば効率が上がると予想した。

我々は1つの論文を1枚のスライドにまとめるといった論文のまとめ方に着目した。これは各所で行われているサーベイのまとめ方の1つである。例えば落合陽一氏の講義 [5] や濱川研究室での活動等でも行われている。我々はこのまとめ方を参考にし、論文のデータから要点をまとめ

¹ 中京大学 工学研究科 情報工学専攻

² 中京大学 工学部 情報工学科

^{a)} k.tanny.uni@gmail.com

た1枚スライドとして自動生成するシステムによってサーベイの効率化を図ることができると考え、「PTOS」を開発した。PTOSは「Paper To One Slide」の頭文字を取ったシステム名である。

2. 関連研究

2.1 サーベイに関する研究

論文サーベイに関する研究としてサーベイを支援する研究がある[6][7][8][9][10]。サーベイ手法としてWeb上での検索があるが、ユーザが今まで扱っていなかった論文を調べようとするときには問題点がある。第一に、論文サーベイにおいて論文に記載されている参考文献は重要な情報であるが、ある論文の参考文献から遡ることで論文をサーベイしていくには量が非常に膨大な点が挙げられる。第二に、検索ワードを決定することが困難である点が挙げられる。これは論文において研究分野ごとに専門用語が多く存在しているため発生する。第三に、検索によって見つかった論文から目当ての論文を見つける作業が必要となる点が挙げられる。これは、検索によって見つかった論文が必ずしも目当ての論文だけとは限らないからである。伊坂らはユーザの興味をもった論文から参照構造を用いて重要論文の発見を行った[6]。この研究では参考文献の参照構造がWeb上における参照構造に似ていることからWebマイニングのアルゴリズムであるHITSアルゴリズムを使用して重要論文の発見を行っている。宮寺らは論文間の関連を自動抽出しそのデータをもとにして論文関係情報の視覚化を行い、効率的なサーベイ活動を支援しようとした[7]。また、杉山らはサーベイには研究分野の動向を把握する段階とテーマに関連した研究を把握する段階の2つの段階があると定義し、それらの段階を考慮したサーベイ支援のシステムを提案した[8]。さらに杉山らはユーザの検索履歴およびダウンロードした論文の情報を蓄積し、他のユーザが同じキーワードを検索した際に蓄積した情報を反映するシステムを実装している[9]。

サーベイをする際の支援だけでなくサーベイ後の情報整理を支援する研究もある。西村らは研究初心者におけるサーベイの整理支援による研究分野への理解を目的とし、サーベイした情報を整理し、可視化するインタフェースを実装した[10]。

PTOSでは、サーベイの際検索で見つかった論文を1枚スライドにし、ユーザが研究の大まかな内容を把握し、サーベイの効率を向上させることを目的としている。大まかな内容を把握することでテーマに関係のない論文かどうかにもすぐに把握可能となり、目当ての論文を見つけやすくなると考えられる。

2.2 論文からポスターを生成する研究

我々が論文から1枚スライドを自動生成する研究を行っ

ているように、論文から別の作品を生成する研究がある。我々の研究のようにサーベイを支援することを目的とした研究ではないが、Yutingらは論文の文章から重要な箇所をTextRankアルゴリズム[11]を用いて抽出し、さらにそれらをポスターとして自動生成するシステムを開発した[12]。

2.3 論文の要旨を抽出する研究

サーベイを効率化する手法として論文の要旨を生成する研究がある。中須賀らは論文中の各文から特徴量を抽出し、その文がどれだけ要約に使われやすいかをAbstractを参考に学習することで要約を生成した[13]。衣川らはEncoder-Decoder機構を自動要約タスクに適用したモデルをベースに、論文の章構造を活用したモデルを検討した[14]。この研究では論文においてセクションごとの役割が異なっていることを考慮し、セクション単位でRNNをまとめることで要約精度の向上を狙った。また、Shinは学術論文が持つ「序論」「関連研究」「提案手法」「実験結果」「結論」といった典型的なセグメント構造に着目し、論文の章や節、段落を分割し、それぞれのセグメントで重要文を抽出する手法を提案した[15]。

PTOSでは学術論文から1枚スライドを制作した際のテキストデータを学習させることで論文の本文の各文をカテゴリによって分類し、カテゴリ内でもっとも重要な文章を抽出する。

3. 提案手法

PTOSでは論文の本文のテキストデータから1枚のスライドを生成する。1枚のスライドに重要と考えられる文章を詰め込もうと思えば詰め込めるが、これではユーザが読みにくくなってしまい、論文の大まかな流れを把握することを難しくしてしまう。そこで我々は、論文の本文のテキストデータ一文一文に対して

- 「どんな研究か(What)」
- 「新規性はどこか(Contribution)」
- 「技術の核はどこか(Important Method)」
- 「有効性の検証をどう行ったか(Evaluation)」
- 「今後の展望はあるか(Future Work)」

といった5つのカテゴリで分類を行い、分類ごとに数文を抽出しそれらをスライドに反映した。

この分類の仕方は学術論文の様々な読み方[16][17][18]や、落合陽一氏が「先端技術とメディア表現1 #FTMA15」のスライド[5]の65枚目で紹介していた論文をまとめる際のフォーマットを参考にした。[16][17][18]には実際にはもっと多くの読み方が書かれているが、それでは1枚スライドに入りきらないと考えられるため、[5][16][17][18]を吟味し5つのカテゴリで分類することにした。

カテゴリ分類を行うため、PTOSでは機械学習を利用した。人が論文から1枚スライドを生成した際の文章一文一

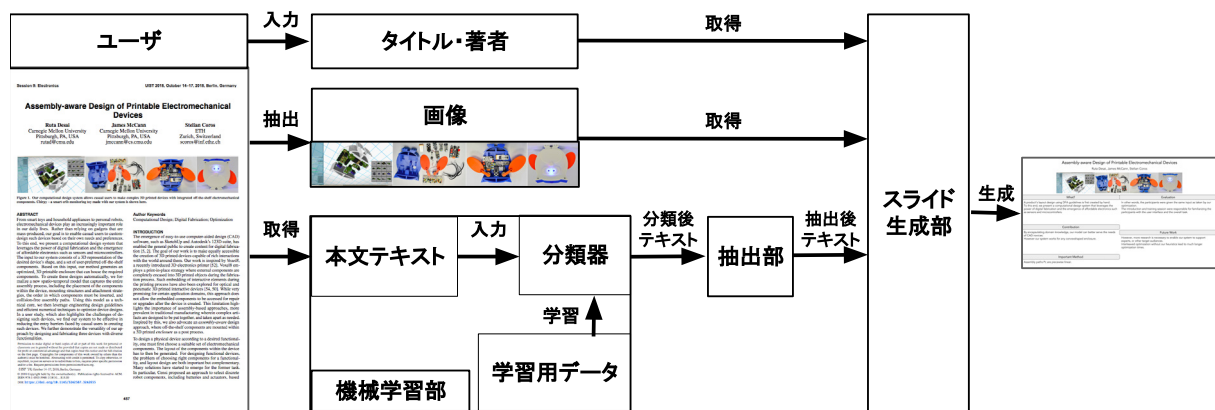


図 1 システム構成

文に対して5つのカテゴリでラベル付けを行い、それを学習に用いることで5クラス分類を行うことのできる分類器を作成する。これによりPTOSでは1枚スライドにした論文本文のテキストデータを入力することで、一文一文を5つのカテゴリに分類することができる。分類したのちスライドに反映する文章を抽出するが、各カテゴリの中で重要な文章を抽出する必要がある。我々は分類器が分類の際に算出するクラス確率が高い文章をカテゴリの中で重要な文章であると定義し、カテゴリごとに確率が高い文章から1~2文を抽出した。そして抽出した文章をもとに1枚スライドを生成するようにしている。また、文章だけでは視覚的にみづらくなると考え、論文のpdfファイルから画像を1枚抽出し、抽出した画像をスライド生成の際に貼り付けるようにした。

4. システム概要

PTOSは大きく分けて機械学習部、抽出部、スライド生成部で成る。図1にシステム構成を示す。以下ではそれぞれの役割について説明する。

4.1 機械学習部

3節でも述べたように、テキストデータ一文一文に対して5クラス分類を行うため、学習データ一文一文に対し5つのラベル「What」「Contribution」「Important Method」「Evaluation」「Future Work」によってラベルづけを行い、学習を行った。テキストの分類を行う場合、SVMやナイーブベイズが主に使われるが今回は多クラス分類問題に対して処理負荷がそれらの手法よりも低いfastTextを利用した。

学習データとして濱川研究室に属する学生が5つのカテゴリをもとにUIST2018の論文から作成した1枚スライド(計78枚)内のテキストデータを使用し、fastTextアルゴリズムにより分類器を作成した。実際に学生によって作成された1枚スライドを図2に示す。学習に使用したデータ

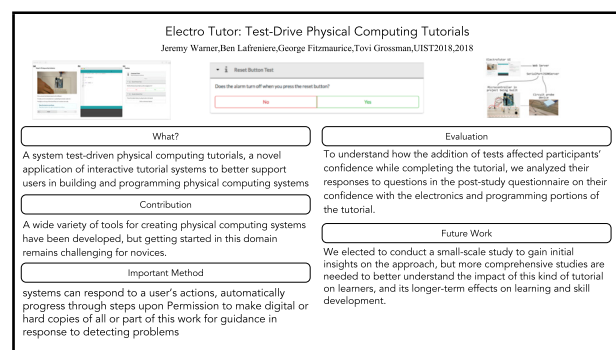


図 2 濱川研究室に属する学生が作成した1枚スライド対象論文[19]

は全て英語で記述されているため、PTOSで1枚スライドを自動生成可能な論文は本文が英語の論文のみに限定される。実際の学習は単語ベクトルの次元数を300、エポック数を500として行った。表1に各ラベルの学習に使用したデータ数を示す。

分類の際の入力データの形式は論文本文データの一文ごとと改行が入った形にした。入力データの作成は論文のpdfファイルからのコピー&ペーストによって行った。これを学習によって作成した分類器に通すことにより一文一文に対して分類されたラベルとその文のクラス確率を算出する。

4.2 抽出部

1枚スライドにした論文のテキストデータを分類器に通すと分類結果とそのクラス確率が算出されるが「分類結果・クラス確率」と「入力された論文のテキストデータ」の対

表 1 学習に使用したデータ

ラベル名	データ数
What	104
Contribution	109
Important Method	93
Evaluation	112
Future Work	90

論文のタイトル	
論文の筆者	
論文の画像	
What	Evaluation
Contribution	
Important Method	Future Work

図 3 1 枚スライドテンプレート

応付けやラベルごとのクラス確率のランク付けなどはできていない。そこで抽出部では機械学習部により算出されたデータの整形を行う。これは Python のライブラリである Pandas を利用した。まず、分類結果を 1 つの csv ファイルとして書き出し、それをプログラム内で読み込む。このデータの形式は二次元配列となる。その後論文のテキストデータを読み込む。これも二次元配列となるため、2 つの配列を結合することで分類結果とクラス確率、論文のテキストデータの対応付けを行う。

データの対応づけを行った後は結合した二次元配列をラベルごとに分割する。これにより 5 つの二次元配列が生成される、これらをそれぞれクラス確率によって降順にソートすることでラベルごとのクラス確率のランク付けを行う。以上でデータの整形が完了となる。

整形後、各ラベルに対しスコアの高い文章から 2 文を抽出するが、ラベル「Important Method」のみ抽出する文章の数を 1 文としている。これはもともとの分類の考えが「技術の核」であるからで 2 文も 3 文も取り出すと「核」となる部分が不明瞭になってしまうと考えたためである。

4.3 スライド生成部

抽出した文章を 1 枚スライドとして自動生成するために PTOS では Google Slides API を使用し、生成した 1 枚スライドを Google Slide として書き出す。これは他者との共有を容易にすることを狙ったからである。

我々は、自動生成の際に必要な要素を以下のように定義した。

- 論文タイトル
- 論文の筆者
- 論文の画像
- 各カテゴリのラベル
- 各カテゴリで抽出した文章

生成の際のテンプレートは文字色や矢印を必要としないシンプルなテンプレートにした。図 3 に実際のテンプレートを示す。

論文タイトルと著者に関してはシステム上でユーザがコマンドライン上で入力することで取得し、スライドに反映

ElectroTutor: Test-Driven Physical Computing Tutorials Jeremy Warner, Ben Lafreniere, George Fitzmaurice, Tovi Grossman	
	
What?	Evaluation
In this paper, we introduce test-driven physical computing tutorials, a new approach that is inspired by test-driven development practices in software engineering. We then present ElectroTutor, a tutorial system that implements a range of tests for both the software and physical aspects of a physical computing system.	Finally, all participants disagreed with the statement "working through the tests was frustrating". Overall, the results are encouraging, with 5/6 participants indicating strong agreement that the tests increased their confidence that they were doing each step correctly.
Contribution	Future Work
A wide variety of tools for creating physical computing systems have been developed, but getting started in this domain remains challenging for novices. Based on any deviations from expected values, the system can provide feedback on why the test failed, or troubleshooting tips from the tutorial author.	Another consideration is how tests will be authored. Adding more sophisticated hardware sensing is another interesting direction for future work.
Important Method	
Voltage measurements are performed using the analogRead function from the Arduino internal library.	

図 4 生成結果：対象論文 [19]

し、抽出部によって抽出したテキストデータは各カテゴリの文章の箇所に反映した。画像の抽出には Poppler を使用した。Poppler は pdf ファイルの閲覧等に用いられるライブラリであり、Poppler の機能の中に pdf ファイルから画像を抽出する機能があるため、それを使用することで pdf ファイル画像を抽出した。この機能は pdf ファイル内の画像を全て抽出するが、論文によっては大量の画像があると予想され、全てをスライドに反映してしまうと 1 枚スライドのレイアウトが崩れる可能性がある。この問題を回避するため、スライド反映するのは抽出した画像の 1 枚目のみとした。また、レイアウト崩れる可能性があるもう 1 つの要素として抽出した画像のサイズがある。この問題を解決するため Python の Python Imaging Library を用いてリサイズ処理を行った。リサイズによって画像をスライドのレイアウトに適したサイズにしスライドへと反映するようにした。図 4 に実際に生成されたスライドの例を示す。自動生成に使用した論文は UIST2018 の「ElectroTutor: Test-Driven Physical Computing Tutorials」[19] である。

5. 評価

PTOS の有効性を検証するために以下の評価を行った。

- 手動作成との比較
- 性能評価

5.1 手動作成との比較

サーベいの効率化には論文 1 つ 1 つの大まかな流れを把握する時間が短ければ短いほど良い。そこで PTOS における 1 枚スライドの生成時間が人よりも短かければ PTOS がサーベいの効率化に有効であると考えた。そのため、人との比較実験として次の実験を行った。

5.1.1 手順

- (1) 濱川研究室内で開催された UIST2018 論文輪読会における一人一人の一枚スライドを作成する時間を計測。(計 78 枚)
- (2) PTOS で UIST2018 の論文輪読会で紹介された全論文から一枚スライドを作成する時間を計測。(計 78 枚)
- (3) 1 と 2 それぞれの合計時間から論文 1 つあたりのスラ

イド作成時間を算出し、比較。

実際に紹介された論文の平均ページ数は 11.5 ページ、最大は 17、最小で 7 ページであった。また、PTOS でのスライド生成は表 2 に示した計算機環境で行った。

5.1.2 結果

論文 1 つあたりのスライド作成時間の平均は表 3 のようになった。PTOS は作成時間において、人よりも 2 時間 10 分 8 秒早く論文 1 つからスライドを作成できることがわかった。よって、PTOS は論文 1 つ 1 つに対してのサーベイ時間を大幅に短縮可能にし、実際のサーベイの効率を向上できると考えられる。

5.2 性能評価

5.1 節にて時間の点からサーベイが効率化できていることはわかったが、分類され抽出された文章が分類されたカテゴリの文章として妥当かどうかといったシステムの性能面の評価も行う必要がある。そのため、以下の実験を行った。

5.2.1 手順

- (1) UIST2018 論文輪読会のスライドから 3 枚を抜粋
対象論文 [19][20][21]
- (2) PTOS で作成した UIST2018 の全論文の 1 枚スライドから 1 と同じ論文のスライドを抜粋
- (3) 被験者が 3 × 2 枚のスライドについて、各項目「What」「Contribution」「Important Method」「Evaluation」「Future Work」の文章として適切な文章かどうかを 1 (全く適していない) ～5 (きわめて適している) で点数付け

本実験は濱川研究室に属する学生 11 人に依頼した。実験の際、スライドの作成者が PTOS か人かは伏せて行った。

5.2.2 結果

各論文に対してのスコア付けの結果の平均を表 4 に示す。表では 5 つのカテゴリを省略して「What」「Cont (Contribution)」「Method (Important Method)」「Eval (Evaluation)」「Future (Future Work)」として表記している。

論文 1 について、「Important Method」「Future Work」のスコアは人が PTOS に大差をつけているが、他の項目ではそこまでの差はなく、「What」の項目では PTOS の方が高い結果となった。また、論文 3 については人との差

表 2 評価実験での計算機環境

デバイス	MacBook Air (13-inch, Mid 2013)
OS	macOS Sierra
CPU	1.3GHz Dual Core i5
メモリ	4GB 1600MHz DDR3

表 3 論文 1 つあたりのスライド作成時間の平均

	人	PTOS
作成時間	2 時間 10 分 41 秒	33 秒

が全項目で小さく、「Contribution」「Important Method」「Evaluation」の項目で人よりも PTOS の方が高い結果が得られた。したがって PTOS は人が作成したスライドに近いスライドを生成できると考えられる。

しかしながら、論文 2 については、PTOS のスコアは全項目低い値となっており、項目として妥当でない文を抽出することがあると考えられる。こういった結果につながった理由は学習データの不足や学習データが妥当でないといった点が考えられる。

5.2.3 アンケート

5.2 節の実験と並行してアンケートを実施した。アンケート項目は以下のとおりである。

- PTOS を使用したいか
- AI によって作成されたスライドはどれか
- スライドで気になることはあったか

「PTOS を使用したいか」項目については 11 人中 11 人が使用したいといった回答であった。理由としては、「論文の概要を最初から理解しているのとそうでないのでは、論文を読み進める速度に違いが出るだろうと思う為」「サーベイが楽になる」といった意見が多かった。

「AI によって作成されたスライドはどれか」とは 5.2 におけるスライドの作成者が PTOS であるスライドを答える設問である。論文 1 と論文 2 の正答率は高かったが、論文 3 に関しては他 2 つの論文よりも間違える人が多く、人が作成したスライドに近いスライドを生成できたと考えられる。

「スライドで気になることはあったか」については、「Important Method が短くてよくわからない」と回答が得られた。これは抽出する文章の数を 1 つにしたのが裏目に出たと考えられる。今後は Important Method も抽出する文章の数を増やす必要がある。

6. 結論・今後の展望

我々は論文のテキストデータから 1 枚のスライドを生成し、論文サーベイを支援するシステム PTOS を開発した。PTOS はサーベイの効率を向上させることを目的としたシステムであり、実際に評価実験によって、人が作成した時に近いスライドを生成することができた。

しかしながら、課題も存在する。第一に、入力データの生成である。PTOS を使用する際に、ユーザが 1 枚スライ

表 4 各論文のスライドのスコア平均

	What	Cont	Method	Eval	Future
論文 1 : PTOS	4.6	3.7	2.6	3.5	3.2
論文 1 : 人	4.5	4.0	3.9	3.5	3.8
論文 2 : PTOS	2.0	3.1	3.1	3.5	2.5
論文 2 : 人	4.5	3.6	4.2	4.5	4.2
論文 3 : PTOS	4.0	3.9	3.9	3.9	3.8
論文 3 : 人	4.3	3.4	3.6	3.8	4.4

ドにしたい論文から本文のテキストデータをコピー&ペーストする労力を取り除くことができればサーベイの効率はより向上すると考えている。pdf からテキストを抽出するツールや OCR による文字起こしといった手法もあるが、

- 論文が 1 段組か 2 段組かあるいはそれ以外か
- 数式の扱いをどうするか
- 本文以外の余計な要素をいかにして取り除くか

などといった様々な問題を解決しなければならない。これらの問題を解決する手法については今後も検討していく。第二に、5.2 節で行われた実験でも結果として現れていたが文章の妥当性が問題として挙げられる。考えられる原因は学習データが少ない、または学習データとして利用した、学生がサーベイによって作成した文章自体がその項目に適当な文章でなかったことが考えられる。これは、学習データにある文章についての妥当性を評価し、さらにデータ数を増やしていくといったことを今後行っていく必要があると考えられる。また、“However”から始まる文章を抜き出してしまい前後関係がわからなくなるといったケースもあった。今後、PTOS ではキーワードのみを抽出しそれをもとに文章を生成したり、そもそも前後関係が必要となる文を抽出した場合は関わりのある文章を全てスライドに反映したり、分類する対象から除外するといった処理が必要であると考えられる。

PTOS がより良いシステムになっていけばさらなる拡張が期待できる。例えば、従来のキーワードによる検索手法とは違った観点からの検索である。今までに生成したスライドをそれぞれのカテゴリごとで類似度によるネットワークを作成すれば、What の内容による検索や Important Method の内容による検索といったことができる。これによりさらなるサーベイの効率の向上が期待できる。他にも論文同士の項目をもとに類似度を測定することで類似した論文をまとめることができれば目当ての論文を 1 つ見つけたら類似した論文を優先して見つけることができるようになることも期待できる。

参考文献

- [1] Google Scholar トップページ
<https://scholar.google.co.jp/>, 2018 年 12 月 20 日アクセス
- [2] ACM Digital Library トップページ
<https://dl.acm.org/>, 2018 年 12 月 20 日アクセス
- [3] 2017 年のディープラーニング論文 100 選
<https://qiita.com/sakaiakira/items/f225b670bea6d851c7ea>, 2018 年 12 月 25 日アクセス
- [4] Awesome Deep Vision
<https://github.com/kjw0612/awesome-deep-vision>, 2018 年 12 月 25 日アクセス
- [5] 先端技術とメディア表現 1 #FTMA15 スライド
<https://www.slideshare.net/0chyai/1-ftma15>, 2018 年 11 月 20 日アクセス

- [6] 井坂 徳恭, 藤本 敬介, 中山 泰一, 2009, “参照構造を用いた重要論文検索システム”, 研究報告コンピュータと教育 (CE), 2009-CE-99(4), pp.1-8
- [7] 宮寺 庸造, 田地 晶, 及部 佳代子, 2004, “学術論文関係情報のグラフ描画問題に基づく視覚化手法”, 電子情報通信学会論文誌. D-1, 情報・システム. 1, 情報処理 / 電子情報通信学会 編, 87(4), pp.398-415
- [8] 杉山 暢彦, 澤本 潤, 杉野 栄二, 瀬川 典久, 2014, “他者のサーベイ行動を利用した論文検索システムの提案”
<http://www.wiss.org/WISS2014Proceedings/demo/094.pdf>, 2018 年 12 月 20 日アクセス
- [9] 杉山 暢彦, 澤本 潤, 杉野 栄二, 瀬川 典久, 2014, “他者の検索履歴を利用した研究者のための論文検索システムの提案”, 研究報告ヒューマンコンピュータインタラクション (HCI), 2014(11), pp.1-6
- [10] 西村 勇哉, 大杉 隆文, 盛山 将広, 内藤 峻, 松下 光範, 2017, “研究初心者のサーベイ行為を対象とした論文整理支援システムの基礎検討”, <http://must.c.u-tokyo.ac.jp/sigam/sigam15/sigam1504.pdf>, 2018 年 12 月 20 日アクセス
- [11] Rada Mihalcea, Paul Tarau, 2004, “TextRank: Bringing Order into Texts”
<https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf>, 2018 年 12 月 20 日アクセス
- [12] Yuting Qiang, Yanwei Fu, Xiao Yu, Yanwen Guo, Zhi-Hua Zhou, Leonid Sigal, 2017, “Learning to Generate Posters of Scientific Papers by Probabilistic Graphical Models”, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp.51-57.
- [13] 中須賀 謙吾, 鶴岡 慶雅, 2015, “談話構造を利用した学術論文の自動要約生成”, 言語処理学会 第 23 回年次大会 発表論文集, pp.150-153
- [14] 衣川 和亮, 鶴岡 慶雅, 2015, “学術論文の章構造に基づくニューラル自動要約モデル”, 言語処理学会 第 21 回年次大会 発表論文集, pp.569-572
- [15] 辛 夏, 2017, “セグメント構造に基づく学術論文の自動要約”
<https://dSPACE.jaist.ac.jp/dSPACE/bitstream/10119/14148/5/paper.pdf>, 2018 年 12 月 20 日アクセス
- [16] How to read a research paper
<https://ccc.inaoep.mx/~esucar/Clases-semdr/Lecturas/reading.pdf>, 2018 年 12 月 15 日アクセス
- [17] Four tips for reading a scientific paper
<https://www.sciencenewsforstudents.org/blog/eureka-lab/four-tips-reading-scientific-paper>, 2018 年 12 月 15 日アクセス
- [18] How to Read an Engineering Research Paper
<https://cseweb.ucsd.edu/~wgg/CSE210/howtoread.html>, 2018 年 12 月 15 日アクセス
- [19] Warner, Jeremy and Lafreniere, Ben and Fitzmaurice, George and Grossman, Tovi, 2018, “ElectroTutor: Test-Driven Physical Computing Tutorials”, Proceeding UIST '18 Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology, pp.435-446.
- [20] Son, Bukun and Park, Jaeyoung, 2018, “Haptic Feedback to the Palm and Fingers for Improved Tactile Perception of Large Objects”, Proceeding UIST '18 Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology, pp.757-763.
- [21] Teyssier, Marc and Bailly, Gilles and Pelachaud, Catherine and Lecolinet, Eric, 2018, “MobiLimb: Augmenting Mobile Devices with a Robotic Limb”, Proceeding UIST '18 Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology, pp.53-63.