JISKa, Vol. 3, No. 2, September, 2018, Pp. 110 - 118
ISSN : 2527 – 5836 (print) | 2528 – 0074 (online)

Implementation of Cosine Similarity in an automatic classifier for comments

Muhammad Habibi[(1)] Sumarsono[(2)]
Program Studi Teknik Informatika [(1) (2)]
Universitas Jenderal Achmad Yani [(1)]
Universitas Islam Negeri Sunan Kalijaga[(2)]
e-mail : muhammadhabibi17@gmail.com [(1)], sumarsono@uin-suka.ac.id [(2)]

Abstract

Classification of text with a large amount is needed to extract the information contained in it. Student comments containing suggestions and criticisms about the lecturer and the lecture process on the learning evaluation system are not well classified, resulting in a difficult assessment process. So from that, we need a classification model that can classify comments automatically into classification categories. The method used is the Cosine Similarity method, which is a method for calculating similarities between two objects expressed in two vectors. The data used in this study were 1,630 comment data with several different categories. The test in this study uses k-fold cross-validation with k = 10. The results showed that the percentage accuracy of the classification model was 80.87%.

Keywords : Classification, Cosine Similarity, Text Mining

Abstrak

Pengklasifikasian teks dengan jumlah yang besar diperlukan untuk mengektraksi informasi yang terkandung dalamnya. Komentar mahasiswa yang berisi saran dan kritik mengenai dosen dan proses perkuliahan pada sistem evaluasi pembelajaran tidak terklasifikasi dengan baik, sehingga mengakibatkan proses penilaian menjadi sulit. Maka dari itu diperlukan suatu model klasifikasi yang dapat mengklasifikasikan komentar secara otomatis ke dalam kategori klasifikasi. Metode yang digunakan adalah metode *Cosine Similarity*, yang merupakan metode untuk menghitung kesamaan antara dua buah objek yang dinyatakan dalam dua buah *vector*. Data yang digunakan dalam penelitian ini sebanyak 1.630 data komentar dengan beberapa kategori berbeda. Pengujian pada penelitian ini menggunakan *k-fold cross validation* dengan k=10. Hasil penelitian menunjukkan bahwa presentase akurasi model klasifikasi adalah sebesasr 80,87%.

Kata Kunci : Klasifikasi, Cosine Similarity, Text Mining

1. PENDAHULUAN

Perkembangan teknologi pada era sekarang ini memiliki dampak yang sangat signifikan dalam kehidupan manusia, mulai dari kehidupan sosial, budaya, ekonomi, bahkan pendidikan. Dalam dunia Pendidikan, teknologi mempunyai peranan sangat penting untuk menunjang keberhasilan tujuan dari Pendidikan tersebut. Banyak kegiatan dalam dunia Pendidikan yang membutuhkan keberaadaan teknologi, mulai dari kegiatan sederhana sampai kegiatan yang membutuhkan tingkat kerumitan yang tinggi. Salah satu bentuk kegiatan pokok dalam dunia Pendidikan yang membutuhkan peranan teknologi adalah proses kegiatan belajar mengajar.

Keberhasilan proses belajar mengajar dalam dunia pendidikan dipengaruhi oleh beberapa faktor yaitu tujuan pembelajaran, bahan ajar yang digunakan, kegiatan belajar mengajar, metode, alat, sumber dan evaluasi proses belajar mengajar (Djamarah & Zain, 2016). Adapun pendapat lain, faktor-faktor keberhasilan proses belajar mengajar adalah mahasiswa, dosen, tujuan belajar, materi pelajaran, sarana belajar, interaksi antara mahasiswa dan materi, interaksi antara dosen dan mahasiswa, interaksi antara mahasiswa dan mahasiswa dan lingkungan belajar (Margono, 2003).

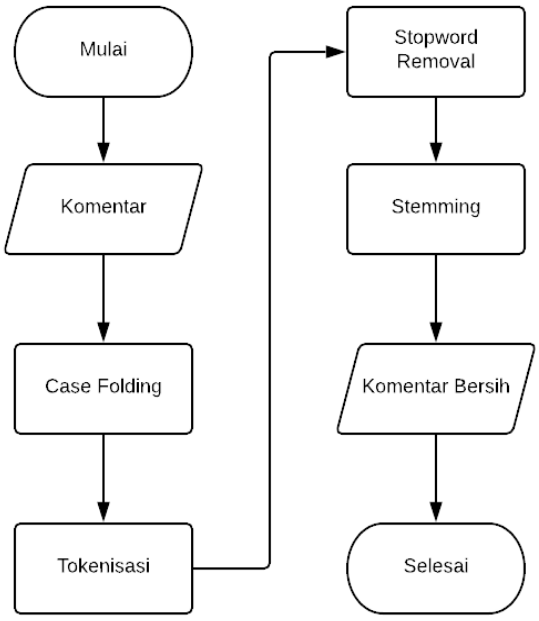
Tingkat kesuksesan pembelajaran pada perguruan tinggi dapat diukur dari opini mahasiswa tentang proses pembelajaran. Opini atau komentar mahasiswa mengenai proses pembelajaran biasanya disimpan pada sistem evaluasi pembelajaran, yaitu dimana mahasiswa memberikan penilaian kepada dosen matakuliah sesuai dengan kinerja dosen saat mengajar. Komentar mahasiswa biasanya berisi saran dan kritik mengenai dosen dan proses perkuliahan (Habibi, 2017). Untuk mengetahui hasil evaluasi tersebut, maka komentar harus dianalisis sehingga dapat diketahui komentar tersebut terkait dengan masalah apa. Proses klasifikasi komentar mahasiswa tersebut biasanya dilakukan dengan cara manual, sehingga membutuhkan usaha ekstra dan waktu yang cukup lama.

Pengklasifikasian teks dengan jumlah yang besar diperlukan untuk mengektraksi informasi yang terkandung dalam kumpulan data teks tersebut. Pengklasifikasian teks tersebut dilakukan dalam upaya untukk memisahkan atau mengelompokkan ciri-ciri atau kategori tertentu. Salah satu metode yang paling sering digunakan dalam pengklasifikasian teks adalah cosine similarity. *Cosine similarity* telah banyak digunakan untuk melakukan pengklasifikasian teks seperti pengklasifikasian tweet populer (Ahmed, Razzaq, & Qamar, 2013), pengklasifikasian pertanyaan ujian (Jayakodi, Bandara, & Meedeniya, 2016), pengklasifikasian jawaban ujian (Saipech & Seresangtakul, 2018) serta untuk pengklasifikasian dokumen text (Kadhim, Cheah, Ahamed, & Salman, 2014).

Tujuan penelitian ini adalah membuat sebuah model klasifikasi yang dapat mengklasifikasikan komentar secara otomatis menggunakan algoritma *cosine similarity* dalam proses pengklasifikasiannya dan menggunakan metode pembobotan TF-IDF. Objek penelitian ini adalah komentar mahasiswa pada system evaluasi pembelajaran. Komentar mahasiswa akan diklasifikasikan ke dalam beberapa kategori secara otomatis. Sehingga diharapkan model klasifikasi yang dihasilkan pada penelitian ini dapat membantu meringankan kegiatan evaluasi pembelajaran.

2. METODE PENELITIAN

Tahapan proses pada penelitian ini diawali dengan tahap *preprocessing* kemudian dilakukan ektraksi fitur setelah itu dilakukan proses klasifikasi. Flowchart preprocessing dapat dilihat pada Gambar 1.



Gambar 1. Flowchart *preprocessing*.

Tahapan pertama adalah *preprocessing*, *preprocessing* data merupakan proses dimana teks yang akan diklasifikasi dibersihkan dan dipersiapkan terlebih dahulu sebelum teks dianalisis (Haddi, Liu, & Shi, 2013). Adapun tahapan *preprocessing* yang akan dilakukan dalam penelitian ini diantaranya adalah:

a. *Case Folding* yaitu proses untuk mengubah huruf menjadi huruf kecil.
b. *Tokenisasi* yaitu proses untuk membagi teks komentar ke dalam token.
c. Dilakukan perubahan *slang word* yaitu proses merubah atau mencari padanan kata.
d. *Stopword removal* yaitu proses untuk menghilangkan kata yang tidak 'relevan' pada hasil parsing. misalnya: kata 'yang','di', dll.
e. *Stemming* yaitu proses mengubah kata berimbuhan menjadi kata dasar. Misalnya: Penilaian = nilai. (*stemming* menggunakan Sastrawi *stemming*)

Pada tahapan *preprocessing*, dilakukan proses standarisasi komentar. Yaitu proses memecah kalimat jika terdapat kata penghubung seperti 'dan', 'tetapi', 'tapi', dan 'namun'. Karena kata penghubung tersebut mengidentifikasikan bahwa dalam satu kalimat terdapat lebih dari satu jenis kategori klasifikasi.

Contoh: cara mengajar bapak menarik tapi sering datang terlambat.
cara mengajar ketepatan waktu

Berdasarkan contoh kalimat di atas, dapat diidentifikasi bahwa dalam satu kalimat terdapat dua kategori klasifikasi. Sehingga kalimat tersebut dilakukan pemecahan untuk mendapatkan hasil yang lebih baik. Berikut merupakan contoh komentar mentah, seperti yang terlihat pada Tabel 1.

Tabel 1. Contoh komentar

Dokumen	Komentar
D1	Cara mengajar bapak menarik tapi sering datang terlambat
D2	Cara menilai bapak subjektif
D3	Kelas selesai tepat waktu namun penilaian harap jangan subjektif
D4	Tugas kuliah terlalu bnyk

Komentar mentah pada Tabel 1 kemudian dilakukan proses *preprocessing*. Selain itu juga dilakukan proses standardisasi komentar, yaitu memecah komentar. Pada komentar D1 dan D3 terdapat kata "tapi" dan "namun" yang ada dalam daftar pemecahan kata, sehingga komentar harus dipecah menjadi dua, yaitu:

- Cara mengajar bapak menarik tapi sering datang terlambat
  - Cara mengajar bapak menarik
  - Sering datang terlambat
- Kelas selesai tepat waktu namun penilaian harap jangan subjektif
  - Kelas selesai tepat waktu
  - Penilaian harap jangan subjektif

Hasil *preprocessing* menghasilkan 6 komentar, seperti yang terlihat pada Tabel 2.

Tabel 2. Komentar hasil *preprocessing*

Dokumen	Komentar
D1	cara ajar tarik
D2	Sering datang lambat
D3	cara nilai bapak subjektif

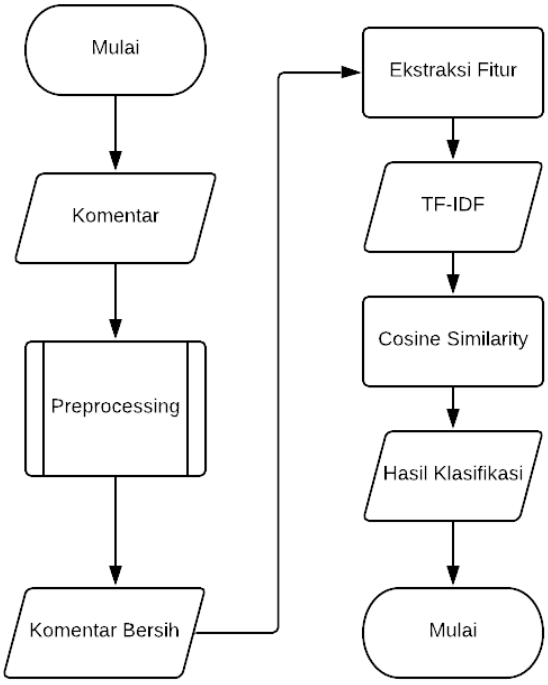
D4	kelas selesai tepat waktu
D5	nilai harap jangan subjektif
D6	tugas terlalu banyak

Berikut merupakan contoh komentar bersih beserta kategori klasifikasi yang dilakukan dengan cara pelabelan manual. Pelabelan manual digunakan untuk proses pengujian, yaitu mencocokkan hasil klasifikasi dengan data label klasifikasi manual. Contoh komentar pelabelan manual dapat dilihat pada Tabel 3.

Tabel 3. Komentar hasil pelabelan manual

Dokumen	Komentar	Klasifikasi
D1	cara ajar tarik	Mengajar
D2	sering datang lambat	Waktu
D3	cara nilai bapak subjektif	Nilai
D4	kelas selesai tepat waktu	Waktu
D5	nilai harap jangan subjektif	Nilai
D6	tugas terlalu banyak	Tugas

Setelah dilakukan *preprocessing*, langkah berikutnya adalah ektraksi fitur yaitu menghitung nilai TF-IDF, kemudian dilakukan klasifikasi menggunakan *cosine similarity*. Flowchart proses klasifikasi dapat dilihat pada Gambar 2.



Gambar 2. Flowchart proses klasifikasi.

Ekstraksi fitur merupakan proses ekstraksi untuk mengidentifikasi entitas-entitas yang dimaksud (Siqueira & Barros, 2010). TF-IDF (*Term Frequency- Inverse Document Frequency)* merupakan *metric* yang umum digunkan dalam proses kategorisasi teks (Sebastiani, 2002). TF-IDF terdiri dari dua buah nilai komponen yaitu *term-frequency* dan *inverse document frequency*. Skema

pembobotan TF-IDF memberikan bobot pada *term t* dalam dokumen *d* yang ditunjukkan oleh persamaan (1) (Manning et al, 2009).

$$tf.idf_{t,d} = tf_{t,d} \times idf_t \quad (1)$$

Nilai $tf_{t,d}$ merupakan bobot suatu *term t* pada dokumen *d* sedangkan idf_t adalah *inverse document frequency* dari *term t*. Persamaan (2) adalah persamaan untuk mencari nilai idf_t. Nilai idf_t diperoleh dari hasil logaritma N dibagi df_t. N merupakan jumlah dokumen keseluruhan dengan df_t adalah banyaknya dokumen yang memuat *term t*.

$$idf_t = \log \frac{N}{df_t} \quad (2)$$

Berdasarkan data komentar pada Tabel 3, didapatkan *corpus* kata unik sebanyak 17 kata unik. Misalkan untuk menghitung kemunculan *term* ‘cara’, maka dilihat berapa kali kemunculan *term* tersebut pada semua dokumen. *Term* ‘cara’ muncul pada dokumen D1 dan D3, *document frequency* (*df*) untuk *term* ‘cara’ adalah 2. Daftar *term* frekuensi dapat dilihat pada Tabel 4.

Tabel 4. *Term* frekuensi

NO	Term	*Term* Frekuensi					
		D1	D2	D3	D4	D5	D6
1	cara	1	0	1	0	0	0
2	ajar	1	0	0	0	0	0
3	tarik	1	0	0	0	0	0
4	sering	0	1	0	0	0	0
5	datang	0	1	0	0	0	0
6	lambat	0	1	0	0	0	0
7	nilai	0	0	1	0	1	0
8	subjektif	0	0	1	0	1	0
9	kelas	0	0	0	1	0	0
10	selesai	0	0	0	1	0	0
11	tepat	0	0	0	1	0	0
12	waktu	0	0	0	1	0	0
13	harap	0	0	0	0	1	0
14	jangan	0	0	0	0	1	0
15	tugas	0	0	0	0	0	1
16	terlalu	0	0	0	0	0	1
17	banyak	0	0	0	0	0	1

Langkah pertama perhitungan TF-IDF adalah menghitung *Inverse document frecuency* (*idf*). Misalnya untuk menghitung nilai *idf term* ‘cara’ yaitu dengan persamaan (2):

$$idf_{(cara)} = \log \frac{N}{df_{(cara)}} = \log \frac{6}{2} = 0.477$$

Setelah *idf* diketahui, maka langkah berikutnya adalah mencari nilai bobot TF-IDF, dengan cara mengalikan nilai *term frequency* dengan nilai *inverse document frequency* untuk masing-masing *term* dengan menggunakan persamaan (1).

Misalkan untuk *term* 'cara' perhitungan TF-IDF nya adalah:

D1 : $tf.idf_{cara,D1} = 1 \times 0.477 = 0.477$
D2 : $tf.idf_{cara,D2} = 0 \times 0.477 = 0$
D3 : $tf.idf_{cara,D3} = 1 \times 0.477 = 0.477$
D4 : $tf.idf_{cara,D4} = 0 \times 0.477 = 0$
D5 : $tf.idf_{cara,D5} = 0 \times 0.477 = 0$
D6 : $tf.idf_{cara,D6} = 0 \times 0.477 = 0$

Untuk perhitungan semua *term* dapat dilihat pada Tabel 5.

Tabel 5. TF-IDF

NO	Term	DF	IDF	TFIDF					
				D1	D2	D3	D4	D5	D6
1	cara	2	0.477	0.477	0	0.477	0	0	0
2	ajar	1	0.778	0.778	0	0	0	0	0
3	tarik	1	0.778	0.778	0	0	0	0	0
4	sering	1	0.778	0	0.778	0	0	0	0
5	datang	1	0.778	0	0.778	0	0	0	0
6	lambat	1	0.778	0	0.778	0	0	0	0
7	nilai	2	0.477	0	0	0.477	0	0.477	0
8	subjektif	2	0.477	0	0	0.477	0	0.477	0
9	kelas	1	0.778	0	0	0	0.778	0	0
10	selesai	1	0.778	0	0	0	0.778	0	0
11	tepat	1	0.778	0	0	0	0.778	0	0
12	waktu	1	0.778	0	0	0	0.778	0	0
13	harap	1	0.778	0	0	0	0	0.778	0
14	jangan	1	0.778	0	0	0	0	0.778	0
15	tugas	1	0.778	0	0	0	0	0	0.778
16	terlalu	1	0.778	0	0	0	0	0	0.778
17	banyak	1	0.778	0	0	0	0	0	0.778

Setelah didapatkan fitur TF-IDF, maka akan dilakukan proses klasifikasi menggunakan Cosine Similarity. Pada proses ini setiap data komentar dihitung panjang vektornya kemudian dibandingkan nilai kedekatan vektor tersebut. Untuk menghitung nilai similarity antar komentar menggunakan persamaan (3) (Lops, Gemmis, & Semeraro, 2010).

$$Cos(x,y) = \frac{\sum_{i=1}^{n} a_{xi} b_{yi}}{\sqrt{\sum_{i=1}^{n} a_{xi}^2 \cdot \sum_{i=1}^{n} b_{yi}^2}} \quad (3)$$

Dimana, a_{xi} : term ke-i yang terdapat pada dokumen *x* dan b_{yi}: term ke-i yang terdapat pada dokumen *y*.

3. HASIL DAN PEMBAHASAN

Data komentar yang digunakan dalam penelitian ini sebanyak 1.630 data komentar. Proses evaluasi kinerja *classifier* menggunakan pendekatan *k-fold cross-validation*. Yaitu setiap *record* digunakan beberapa kali dalam jumlah yang sama untuk pelatihan dan pengujian. Pada penelitian ini digunakan *k*=10, pada 10-*fold cross-validation* data akan dibagi menjadi 10 subset dengan ukuran yang sama dan data yang berbeda. Adapun distribusi data komentar secara rinci diperlihatkan pada Tabel 6.

Tabel 6. Distribusi data komentar

Kelas Kategori	Jumlah Komentar
Cara Mengajar	440
Ketepatan Waktu	395
Cara Penilaian	400
Pemberian Tugas	395

Confussion matrix merupakan salah satu *tools* penting dalam metode visualisasi yang digunakan pada mesin pembelajaran yang biasanya memuat dua kategori atau lebih (Manning, Raghavan, & Schutze, 2009). Setiap unsur matriks menunjukkan jumlah contoh data uji untuk kelas sebenarnya yang digambarkan dalam bentuk baris sedangkan kolom menggambarkan kelas yang diprediksi. Berikut merupakan contoh *confusion matrix* prediksi dua kelas seperti yang terlihat pada Tabel 7.

Tabel 7. contoh confusion matrix prediksi dua kelas

	Predicted Class-1	*Predicted Class-2*
Actual Class-1	*True Posititive*	*False Negative*
Actual Class-2	*False Positive*	*True Negative*

Nilai *true positive* (TP) dan *true negative* (TN) adalah hasil klasifikasi yang benar. Nilai *false positive* (FP) adalah nilai dimana hasilnya diprediksi sebagai *class-1* namun sebenarnya merupakan *class-2* sedangkan *false negative* (FN) adalah nilai dimana prediksi mengklasifikasikan *Class*-2 namun faktanya termasuk dalam *class*-1. Nilai akurasi *confusion matrix* berdasarkan Tabel 5 diperoleh dengan persamaan (10).

$$accuray = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

Pengujian akurasi *Cosine Similarity* dilakukan dengan 10 kali pengujian menggunakan *k-fold cross validation*. Dari 10 kali pengujian tersebut didapatkan nilai akurasi rata-rata *Cosine Similarity* sebesar 80,87%. Hasil pengujian akurasi *Cosine Similarity* dapat dilihat pada Tabel 8.

Hasil dari pengujian mengungkapkan bahwa nilai akurasi yang didapatkan dari algoritma cosine similarity memiliki nilai akurasi yang cukup baik. Berdasarkan hasil dari 10 kali percobaan dengan 10 pembagian data set, didapatkan nilai akurasi rata-rata 80,87%. Dari 10 percobaan tersebut, terdapat tiga kali percobaan yang mendapatkan nilai akurasi dibawah 80%. Yaitu percobaan ke-5, ke-7 dan ke-10, dengan masing-masing nilai akurasi yang didapatkan adalah 78,06%, 77,46% dan 76,69%. Hal ini terjadi kemungkinan besar disebabkan karena dalam proses *preprocessing* masih ada beberapa kata yang belum tercakup dalam proses *stemming*, sehingga menyebabkan kata yang seharusnya sama tapi teridentifikasi menjadi kata yang berbeda.

Tabel 8. Pengujian akurasi Cosine Similarity

Pengujian	Akurasi
1	83,23%
2	82,25%
3	80,50%
4	83,73%
5	78,06%
6	81,33%
7	77,46%
8	82,80%
9	82,61%
10	76,69%
Akurasi rata-rata	80,87%

4. KESIMPULAN

Penelitian ini telah berhasil membuat model klasifikasi otomatis untuk mengklasifikasikan komentar dari sistem evaluasi pembelajaran menggunakan *cosine similarity.* Hasil dari percobaan yang dilakukan menunjukkan bahwa model klasifikasi yang telah dibuat memiliki akurasi rata-rata sebesar 80,87%. Tetapi, dalam bererapa kasus, kami memperhatikan nilai akurasi dari model klasifikasi tampak agak rendah, sebagai akibat dari proses *preprocessing.* Untuk meningkatkan kinerja dari model klasifikasi, kami bermaksud untuk mengintegrasikan analisis simantik dan analisis konteks ke dalam penelitian selanjutnya.

DAFTAR PUSTAKA

Ahmed, H., Razzaq, M. A., & Qamar, A. M. (2013). Prediction of popular tweets using Similarity Learning. *ICET 2013 - 2013 IEEE 9th International Conference on Emerging Technologies*. https://doi.org/10.1109/ICET.2013.6743524

Djamarah, & Zain. (2016). *Strategi Belajar Mengajar*. Jakarta: Rineka Cipta.

Habibi, M. (2017). *Analisis Sentimen dan Klasifikasi Komentar Mahasiswa pada Sistem Evaluasi Pembelajaran Menggunakan Kombinasi KNN Berbasis Cosine Similarity dan Supervised Model*. *Departemen Ilmu Komputer dan Elektronika, Fakultas Matematika dan Ilmu Pengetahuan Alam*. Universitas Gadjah Mada.

Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis.

Procedia Computer Science, *17*, 26–32. https://doi.org/10.1016/j.procs.2013.05.005

Jayakodi, K., Bandara, M., & Meedeniya, D. (2016). An automatic classifier for exam questions with WordNet and Cosine similarity. *2nd International Moratuwa Engineering Research Conference, MERCon 2016*, 12–17. https://doi.org/10.1109/MERCon.2016.7480108

Kadhim, A. I., Cheah, Y. N., Ahamed, N. H., & Salman, L. A. (2014). Feature extraction for co-occurrence-based cosine similarity score of text documents. *2014 IEEE Student Conference on Research and Development, SCOReD 2014*, 2–5. https://doi.org/10.1109/SCORED.2014.7072954

Lops, P., Gemmis, M. De, & Semeraro, G. (2010). *Recommender Systems Handbook*. *Recommender Systems Handbook*. https://doi.org/10.1007/978-0-387-85820-3

Manning, C. D., Raghavan, P., & Schutze, H. (2009). *An Introduction to Information Retrieval*. *Information Retrieval*. https://doi.org/10.1109/LPT.2009.2020494

Margono. (2003). *Metode Penelitian Pendidikan*. Jakarta: Rineka Cipta.

Saipech, P., & Seresangtakul, P. (2018). Automatic Thai Subjective Examination using Cosine Similarity. *ICAICTA 2018 - 5th International Conference on Advanced Informatics: Concepts Theory and Applications*, 214–218. https://doi.org/10.1109/ICAICTA.2018.8541276

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, *34*(1), 1–47.

Siqueira, H., & Barros, F. (2010). A Feature Extraction Process for Sentiment Analysis of Opinions on Services. *Proceedings of the III International Workshop on Web and Text Intelligence (WTI)*.