Kelompok : Natural Language Programmer

Anggota :

1. Alif Ahsanil Satria - 1606882540
2. Anindito Bhagawanta Hidayat - 1606879230
3. Aditya Yudha Pratama - 1606917683

BAB I
PENDAHULUAN

1.1 Latar Belakang

Saat ini, penggunaan bahasa Indonesia di internet semakin banyak dan banyak dari kalimat yang ada mengandung kata yang ambigu. Contoh kata yang ambigu adalah kata ‘bulan’ dalam kalimat “Malam ini bulan purnama terlihat indah”, kata ‘bulan’ dalam Bahasa Indonesia memiliki arti satelit dari planet bumi atau bulan sebagai satuan waktu. Dalam *NLP*, task untuk mencari makna yang tepat dari sebuah kata di dalam suatu konteks kalimat disebut *word sense disambiguation* (WSD). Banyak manfaat yang bisa didapat jika kita bisa melakukan task ini dengan baik, seperti menyelesaikan problem *text classification, text clustering,* dan untuk *machine translation*. Pada laporan ini, kami mengajukan penggunaan algoritma SVM untuk menebak makna yang tepat dengan tf-idf sebagai metode *feature extraction* dan file single_annotator, double_annotator_agree, dan triple_annotator_agree sebagai data training.

1.2 Studi Literatur

Untuk studi literatur, kami menggunakan paper yang terdapat pada link https://ieeexplore.ieee.org/document/8549824 (file paper terdapat pada lampiran). Berdasarkan paper tersebut, kami mempelajari bahwa langkah-langkah yang dibutuhkan untuk menyelesaikan task WSD adalah *preprocessing* (*lowercase* semua kata, *punctuation removal, stemming, stopwords removal, normalize slang word, normalize abbreviation word)* untuk mengeliminasi *noise word* pada sentence yang kurang atau tidak memiliki pengaruh dalam penentuan makna yang tepat pada kata tersebut, lalu

dilanjutkan dengan TF-IDF *feature extraction*, dan terakhir menggunakan SVM untuk menebak makna kata yang tepat di dalam konteks kalimat tersebut.

1.3 Rumusan Masalah

a. Bagaimana hasil penggunaan algoritma SVM dan TF-IDF sebagai *feature extraction* dalam menyelesaikan WSD task?

1.4 Tujuan Penelitian

a. Membangun model untuk menyelesaikan WSD task dalam menentukan makna kata yang tepat dalam suatu kalimat menggunakan algoritma SVM dan TF-IDF sebagai *feature extraction*.
b. Mengetahui hasil penggunaan algoritma SVM dan TF-IDF sebagai *feature extraction* dalam menyelesaikan WSD task.

BAB II
METODOLOGI

2.1 Metodologi

Langkah-langkah yang dilakukan sebagai berikut :

1. Preprocessing

Pada tahap ini, kami melakukan *lowercase* semua kata, membuang *punctuation* dan karakter selain A-Z, a-z, dan 0-9, membuang semua karakter berbentuk angka, membuang extra space, dan membuang huruf yang berulang seperti “haii”.

2. TF-IDF Feature Extraction

Setelah *preprocessing*, kami menggunakan metode tf-idf sebagai *feature extraction*. TF-IDF digunakan untuk memberikan *weight* pada masing-masing kata di sebuah sentence yang mana kombinasi weight dari kata-kata tersebut digunakan untuk membedakan label sense dari pasangan kata dan konteks kalimat yang ingin dicari

dengan metode SVM . *Weight* itu sendiri merepresentasikan "importance" dari sebuah kata pada konteks kalimat tersebut dalam menentukan makna yang tepat dari kata yang ingin dicari *sense* nya.

T_k = term k in document D_i
tf_{ik} = frequency of term T_k in document D_i
idf_k = inverse document frequency of term T_k in *Corpus*
N = total number of documents in the collection *Corpus*
n_k = the number of documents in *Corpus* that contain T_k

$$tfidf_k = tf_k \times \log\left(\frac{N}{n_k}\right)$$

Contoh penjelas untuk TF :

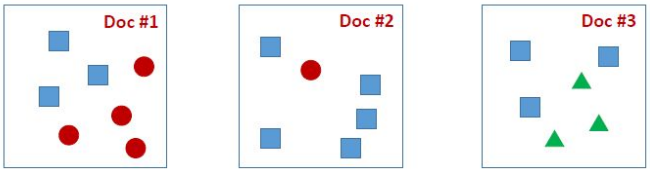
Doc	Content
1	T1 T2 T2 T3
2	T3 T3 T1 T3 T4
3	T2 T3 T4 T2 T2 T5

V = {T1, T2, T3, T4, T5}

	D1	D2	D3
T1	1	1	0
T2	**2**	0	**3**
T3	1	**3**	1
T4	0	1	1
T5	0	0	1

Contoh penjelas untuk IDF :

Our corpus/documen collection:



$$idf(\blacktriangle) = \log(N/n_k) = \log(3/1) = 0.47$$
$$idf(\blacksquare) = \log(N/n_k) = \log(3/3) = 0$$
$$idf(\bullet) = \log(N/n_k) = \log(3/2) = 0.17$$

TF-IDF = TF x IDF

Selain itu, kami menambahkan 1 pendekatan tambahan, yaitu dengan menggunakan *single value decomposition (SVD)* sebagai *dimensionality reduction.* Jadi, hasil *feature*

extraction dari TF-IDF kami gunakan sebagai input untuk SVD dan outputnya dijadikan input untuk SVM. Secara keseluruhan, kami menggunakan 2 pendekatan, yaitu tanpa SVD dan dengan SVD sebagai *dimensionality reduction* dengan menyisakan fitur sebanyak 15 buah dan 20 iterasi.

3. Prediction

Setelah melakukan *feature extraction*, kami menggunakan SVM dengan kernel linear untuk menebak labelnya.

BAB III
ANALISIS

3.1 Eksperimen dan Hasil

Secara ringkas, skenario eksperimen yang dilakukan adalah sebagai berikut :

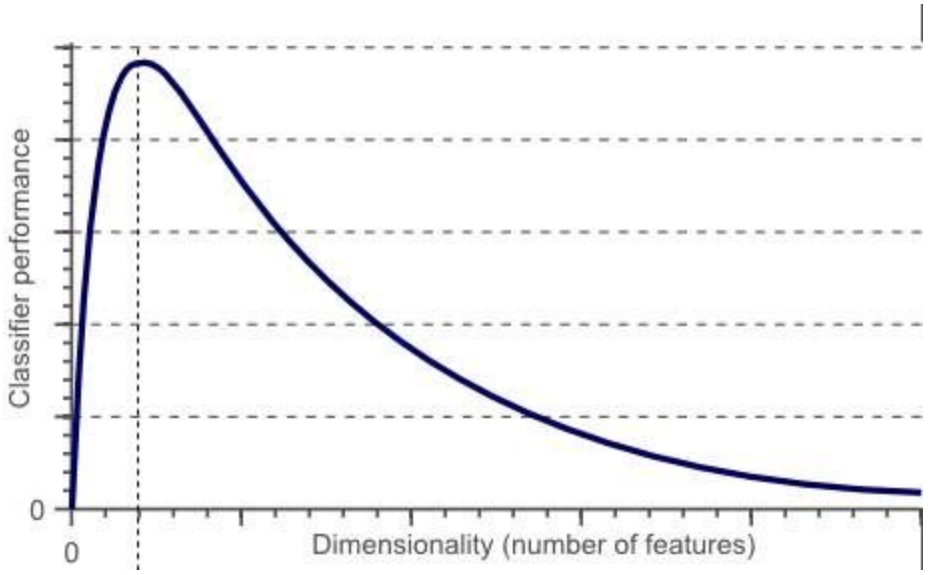
1. Preprocessing → TF-IDF Feature Extraction → SVM
2. Preprocessing → TF-IDF Feature Extraction → SVD Dimensionality Reduction → SVM

Untuk skenario 1, micro-accuracy nya sebesar 57.81 % (seingat kami) dan sisanya sudah lupa karena history submission di web online evaluation nya tidak diperlihatkan.

Untuk skenario 2, hasilnya adalah sebagai berikut :

No	Group	Micro Accuracy (%)	Macro Accuracy (%)	Macro Precision (%)	Macro Recall (%)	Macro F1 (%)
4	Natural Language Programmer	59.41	62.66	52.16	46.88	44.79

Berdasarkan hasil di atas, terlihat bahwa skenario 2 memberikan hasil yang lebih baik daripada skenario 1. Alasannya adalah karena kita melakukan *dimensionality reduction* terlebih dahulu setelah output dari tf-idf didapat dan sebelum dimasukkan ke dalam svm classifier. Hal ini bisa dijelaskan oleh fenomena yang disebut dengan *curse of dimensionality* yang tergambar pada grafik berikut :



Curse of dimensionality menjelaskan fenomena bahwa performa dari suatu *classifier* awalnya meningkat seiring dengan meningkatnya banyaknya fitur sampai ke titik tertentu ketika performa tersebut sudah mencapai puncaknya. Saat itu, ketika jumlah fitur ditambah, maka performanya akan menurun.

BAB IV
PENUTUP

4.1 Kesimpulan

Sejauh ini, kami sudah menyelesaikan task WSD dengan metode TF-IDF *feature extraction* + SVM dan TF-IDF *feature extraction* + SVD *dimensionality reduction* + SVM. Akurasi dari hasil percobaan masih berada di bawah 60%, sehingga belum cukup baik dalam menyelesaikan task WSD ini. Meskipun demikian, setidaknya hasil yang kami dapatkan sudah lebih baik daripada *uniform random chance* karena akurasi yang didapat sudah lebih dari 50% dan tiap kata yang ingin dicari sense nya mempunyai kemungkinan label sense yang bervariasi, ada yg 2,3,4,5,6,7 kemungkinan sense yang ingin ditebak tergantung dari kata nya itu sendiri.

4.2 Saran

Untuk kedepannya, saran penyempurnaan dari kami adalah dengan memperbanyak data training karena metode TF-IDF membutuhkan training data yang banyak supaya representasi fitur nya jadi lebih baik. Selain itu, saran implementasi lainnya dari kami adalah dengan menggunakan pendekatan *knowledge base* supaya hal-hal yang sebelumnya belum pernah digali oleh kami (seperti *gloss* dan isi makna suatu kata) bisa diberdayakan (tidak hanya kata dengan *sense label* nya saja).

Daftar Pustaka

- E. Faisal, F. Nurifan, and R. Sarno, "Word Sense Disambiguation in Bahasa Indonesia Using SVM," *2018 International Seminar on Application for Technology of Information and Communication*, 2018.