

BAB 6. KESIMPULAN DAN SARAN

Setelah selesai melakukan penelitian mengenai analisis sentimen terhadap pandangan masyarakat akan Covid-19 dan pelaksanaan Vaksin Covid-19 di Indonesia dengan menggunakan metode Multinomial Naïve Bayes (MNB), Support Vector Machine (SVM), *Multinomial Naïve Bayes (MNB) feature extraction SentiStrength* dan *Support Vector Machine (SVM) feature extraction SentiStrength*. Peneliti akan masuk ke bagian kesimpulan dan saran terkait hasil dari penelitian tersebut. Berikut ini adalah kesimpulan dan saran yang diberikan oleh peneliti:

6.1. Kesimpulan

Berikut ini adalah kesimpulan dari hasil penelitian yang telah dilakukan di atas:

1. Hasil dari klasifikasi dengan menggunakan metode *Multinomial Naïve Bayes (MNB)*, *Support Vector Machine (SVM)*, *Multinomial Naïve Bayes (MNB) feature extraction SentiStrength* dan *Support Vector Machine (SVM) feature extraction SentiStrength* untuk analisis sentimen terhadap Covid-19 dan Vaksin Covid-19 menunjukkan hasil bahwa metode *Support Vector Machine (SVM) feature extraction SentiStrength* mendapatkan nilai Akurasi yang tertinggi untuk masing-masing analisis. Untuk analisis Covid-19 dengan metode *Support Vector Machine (SVM) feature extraction SentiStrength* mendapatkan nilai Akurasi sebesar 85% dan untuk analisis Vaksin Covid-19 dengan metode *Support Vector Machine (SVM) feature extraction SentiStrength* mendapatkan nilai Akurasi sebesar 81%. Dapat diambil kesimpulan bahwa metode *Support Vector Machine (SVM) feature extraction SentiStrength* untuk kedua analisis tersebut baik dalam melakukan klasifikasi pada data.
2. Metode *Support Vector Machine (SVM)* mendapatkan nilai Akurasi yang tertinggi untuk masing-masing analisis Covid-19 sebesar 81% dan untuk Vaksin Covid-19 sebesar 76%.

3. Hasil dari validasi untuk analisis Covid-19 dan Vaksin Covid-19 dengan metode *Multinomial Naïve Bayes* (MNB), *Support Vector Machine* (SVM), *Multinomial Naïve Bayes* (MNB) *feature extraction SentiStrength* dan *Support Vector Machine* (SVM) *feature extraction SentiStrength* menunjukkan metode *Support Vector Machine* (SVM) *feature extraction SentiStrength* mendapatkan hasil Netral yang tertinggi. Dimana untuk Covid-19 mendapatkan nilai Netral lebih tinggi 90% daripada Vaksin Covid-19 untuk positif sebesar 89%.
4. Persentase dari analisis sentimen untuk Covid-19 dan Vaksin Covid-19 menunjukkan bahwa respon masyarakat akan Covid-19 yang sudah berjalan hampir 2 tahun menunjukkan respon Positif dengan nilai persentase diatas 50%. Sedangkan respon masyarakat untuk pelaksanaan Vaksin yang sudah berjalan 1 tahun lebih menunjukkan respon Positif dengan nilai persentase 44%.
5. Persentase dari analisis sentimen untuk Covid-19 SentiStrength dan Vaksin Covid-19 SentiStrength menunjukkan bahwa respon masyarakat akan Covid-19 yang sudah berjalan hampir 2 tahun menunjukkan respon Netral dengan nilai persentase diatas 50%. Sedangkan respon masyarakat untuk pelaksanaan Vaksin yang sudah berjalan 1 tahun lebih menunjukkan respon Netral dengan nilai persentase diatas 50%.
6. Pandangan masyarakat Indonesia untuk Covid-19 yang sudah ada di Indonesia selama hampir 2 tahun lebih terbilang sudah positif. Masyarakat sudah terbiasa dengan virus tersebut dan masyarakat sudah terbiasa melihat orang ataupun masyarakat yang terkena Covid-19. Bukan seperti awal virus ini muncul masyarakat sangat takut dan bahkan menjauhi jika ada masyarakat lain yang terkena virus Covid-19 tersebut. Dari data penelitian juga menunjukkan bahwa pandangan masyarakat akan covid-19 memperoleh persentase positif diatas 50%. Dan pandangan masyarakat Indonesia di tahun depan akan lebih tinggi di tahun kedepannya.
7. Pandangan masyarakat Indonesia untuk pelaksanaan Vaksin Covid-19 masih cukup netral dikarenakan nilai persentase dari positif dan negatif tidak terlalu

jauh. Dimana positif sebesar 44% dan negatif 30%. Di tahun kedepannya pandangan masyarakat akan vaksin akan meningkat dengan strategi-strategi khusus dari pemerintah untuk membuat masyarakat percaya akan sebaiknya melakukan vaksin covid-19.

6.2. Saran

Saran yang diberikan oleh peneliti adalah sebagai berikut:

1. Analisis untuk padangan Covid sebaiknya tidak perlu dilakukan lagi karena di tahun akan mendatang pandangan masyarakat akan menghasilkan respon positif yang lebih tinggi. Mungkin perlu dilakukannya lagi analisis mengenai pelaksanaan Vaksin di Indonesia di tahun 2022, dimana pelaksanaan vaksin tersebut akan menempuh 50%-70% keseluruhan masyarakat di Indonesia. Analisis tersebut akan terlihat apakah pandangan masyarakat akan mendapatkan respon positif yang lebih tinggi atau tidak.
2. Melakukan analisis sentimen untuk pandangan masyarakat akan jenis-jenis Vaksin yang sudah di sebarakan di Indonesia. Analisis tersebut akan terlihat apakah pandangan masyarakat akan mendapatkan respon positif terhadap jenis Vaksin yang telah mereka terima atau ada penyesalan terhadap jenis Vaksin yang telah mereka terima.
3. Perlu menambahkan beberapa tambahan metode lainnya untuk melakukan perbandingan analisis sentimen seperti menambahkan *feature extraction TF-IDF* pada metode sentimen yang dilakukan.

REFERENSI

- [1] M. C. Dhingra, M. S. Sahoo, P. Y. K. Gupta, and M. A. D. Rai, "Analysis of twitter data for India and China Covid-19 Vaccination," *Sanrachana*, no. March 2020, pp. 1–4, 2021.
- [2] F. F. Rachman and S. Pramana, "Analisis Sentimen Pro dan Kontra Masyarakat Indonesia tentang Vaksin COVID-19 pada Media Sosial Twitter," *Heal. Inf. Manag. J. ISSN*, vol. 8, no. 2, p. 100, 2020.
- [3] C. E. K. Fakta, P. Peristiwa, P. Dunia, M. Khas, and H. O. T. Topics, "Update Jumlah Masyarakat Indonesia Sudah Disuntik Vaksin Covid-19 Per 30 Juni Hot Issue Rekomenda," *Merdeka.com*, 2021. .
- [4] Satgas Covid-19 RI, "Lebih dari 69 Juta Penduduk Indonesia Sudah Vaksinasi Covid-19," *Katadata.com*, no. September, p. 2021, 2021, [Online]. Available: <https://databoks.katadata.co.id/datapublish/2021/09/08/lebih-dari-69-juta-penduduk-indonesia-sudah-vaksinasi-covid-19>.
- [5] A. Dwiyanto, "Ada 7 Jenis Vaksin COVID yang Digunakan di Indonesia, Sudah Tersertifikasi WHO?," *DetikHealth*, 2021. .
- [6] J. Yu and J. Muñoz-Justicia, "A bibliometric overview of twitter-related studies indexed in web of science," *Futur. Internet*, vol. 12, no. 5, pp. 1–3, 2020, doi: 10.3390/FI12050091.
- [7] J. Al-kassmy, J. Pedersen, and G. Kobinger, "Vaccine Candidates against Coronavirus Infections .," *Viruses*, vol. 12, no. 861, pp. 1–18, 2020, [Online]. Available: <https://doi.org/10.3390/v12080861>.
- [8] Kementerian Kesehatan Republik Indonesia, "Question (Faq) Pelaksanaan Vaksinasi Covid-," 2020, 2021. .
- [9] M. Popular, "Vaksin COVID-19 Mana yang Paling ' Laku ' di Dunia ? Ini Daftarnya," *Detik Health*, 2021. .

- [10] R. Palacios *et al.*, “Double-Blind, Randomized, Placebo-Controlled Phase III Clinical Trial to Evaluate the Efficacy and Safety of treating Healthcare Professionals with the Adsorbed COVID-19 (Inactivated) Vaccine Manufactured by Sinovac – PROFISCOV: A structured summary of a ,” *Trials*, vol. 21, no. 1. 2020, doi: 10.1186/s13063-020-04775-4.
- [11] L. M. Hampton, R. Aggarwal, S. J. W. Evans, and B. Law, “General determination of causation between Covid-19 vaccines and possible adverse events,” *Vaccine*, vol. 39, no. 10, pp. 1478–1480, 2021, doi: 10.1016/j.vaccine.2021.01.057.
- [12] R. L. Weintraub, L. Subramanian, A. Karlage, I. Ahmad, and J. Rosenberg, “Covid-19 vaccine to vaccination: Why leaders must invest in delivery strategies now,” *Health Aff.*, vol. 40, no. 1, pp. 33–41, 2021, doi: 10.1377/hlthaff.2020.01523.
- [13] M. Müller and M. Salathé, “Addressing machine learning concept drift reveals declining vaccine sentiment during the COVID-19 pandemic,” *arXiv*, vol. 12, no. 10, pp. 1–12, 2020.
- [14] A. S. M. A. Haque, M. Thakur, M. Bielskas, and A. Marathe, “Persistence of anti-vaccine sentiment in social networks through strategic interactions,” *Computational-Epidemiology*, vol. 9, pp. 1–3, 2021.
- [15] F. J. Damanik and D. B. Setyohadi, “Analysis of public sentiment about COVID-19 in Indonesia on Twitter using multinomial naive bayes and support vector machine,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 704, no. 1, 2021, doi: 10.1088/1755-1315/704/1/012027.
- [16] R. Sianipar and E. B. Setiawan, “Pendeteksian Kekuatan Sentimen Pada Teks Tweet Berbahasa Indonesia Menggunakan Sentistrength,” *e-Proceeding Eng. Vol.2, No.3 Desember 2015*, vol. 2, no. 3, pp. 7922–7928, 2015, [Online]. Available: https://openlibrary.telkomuniversity.ac.id/pustaka/files/104179/jurnal_eproc/jurnal_eproc.pdf.

- [17] S. Minaee, E. Azimi, and A. Abdolrashidi, "Deep-Sentiment: Sentiment Analysis Using Ensemble of CNN and Bi-LSTM Models," *Cornel Univ.*, vol. 1, no. 1, pp. 1–3, 2019.
- [18] S. Das, R. K. Behera, M. Kumar, and S. K. Rath, "Real-Time Sentiment Analysis of Twitter Streaming data for Stock Prediction," *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 956–964, 2018, doi: 10.1016/j.procs.2018.05.111.
- [19] Pristiyono, M. Ritonga, M. A. Al Ihsan, A. Anjar, and F. H. Rambe, "Sentiment analysis of COVID-19 vaccine in Indonesia using Naïve Bayes Algorithm," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1088, no. 1, p. 012045, 2021, doi: 10.1088/1757-899x/1088/1/012045.
- [20] M. A. Mohammed Alhajji , Abdullah Al Khalifah , Mohammed Aljubran, "Sentiment analysis of tweets in Saudi Arabia regarding governmental preventive measures to contain COVID-19," *Preprints*, vol. 17, no. 2, pp. 3–10, 2017, doi: 10.20944/preprints202004.0031.v1.
- [21] M. R, B. A, and S. K, "COVID-19 Outbreak: Tweet based Analysis and Visualization towards the Influence of Coronavirus in the World," *Gedrag Organ. Rev.*, vol. 33, no. 02, pp. 534–549, 2020, doi: 10.37896/gor33.02/062.
- [22] A. M. U. D. Khanday, Q. R. Khan, and S. T. Rabani, "Identifying propaganda from online social networks during COVID-19 using machine learning techniques," *Int. J. Inf. Technol.*, vol. 13, no. 1, pp. 115–122, 2021, doi: 10.1007/s41870-020-00550-5.
- [23] P. H. Prastyo, A. S. Sumi, A. W. Dian, and A. E. Permanasari, "Tweets Responding to the Indonesian Government's Handling of COVID-19: Sentiment Analysis Using SVM with Normalized Poly Kernel," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 6, no. 2, p. 112, 2020, doi: 10.20473/jisebi.6.2.112-122.

- [24] B. P. Pokharel, "Twitter Sentiment Analysis During Covid-19 Outbreak in Nepal," *SSRN Nepal Open Univ.*, vol. 1, pp. 1–21, 2020, [Online]. Available: <http://dx.doi.org/10.2139/ssrn.3624719>.
- [25] B. Pratama *et al.*, "Sentiment Analysis of the Indonesian Police Mobile Brigade Corps Based on Twitter Posts Using the SVM and NB Methods," *J. Phys. Conf. Ser.*, vol. 1201, no. 1, pp. 1–3, 2019, doi: 10.1088/1742-6596/1201/1/012038.
- [26] M. Roque López, Javier Tejada, "ITHEA SPANISH SENTISTRENGTH AS A TOOL FOR OPINION MINING PERUVIAN FACEBOOK AND TWITTER Roque López , Javier Tejada , MikeThelwall SentiStrength The proposed classifier," *ITHEA*, vol. 82–85, no. 82, pp. 82–85, 2012, [Online]. Available: http://webcache.googleusercontent.com/search?q=cache:7PgeOfvdcfoJ:roquelopez.com/resource/publications/INFOS2012_Lopez.pdf+&cd=2&hl=id&ct=clnk&gl=id.
- [27] T. Fukuhara, H. Nakagawa, and T. Nishida, "Understanding sentiment of people from news articles: Temporal sentiment analysis of social events," *ICWSM 2007 - Int. Conf. Weblogs Soc. Media*, vol. II, no. 2, pp. 1–2, 2007.
- [28] R. S. Wahono, "Proses Data Mining," *Data Min.*, vol. 2022, no. 1996, pp. 1–143, 2015, [Online]. Available: <http://romisatriawahono.net/dm>.
- [29] M. A. Long, P. B. Stretesky, G. Harms, R. Sollund, P. Macmillan, and B. Chapter, "Northumbria Research Link (www.northumbria.ac.uk/nrl)," *2019 IEEE Glob. Eng. Educ. Conf.*, vol. 51, no. September, pp. 1–51, 2015.
- [30] D. Sentistrength, "SentiStrength," pp. 1–8.
- [31] Sartini, *Analisis Sentimen Twitter Bahasa Indonesia Menggunakan Algoritma Convolutional Neural Network*. 2020.

LAMPIRAN

Library Untuk Mengambil Labelling Nilai SentiStrength:

```
# coding: utf-8

import re
from collections import OrderedDict
import numpy as np

class sentistrength:
    def __init__(self, config=dict()):
        self.negasi = [line.replace('\n','') for line in
open("negatingword.txt").read().splitlines()]
        self.tanya = [line.replace('\n','') for line in
open("questionword.txt").read().splitlines()]
        #create sentiment words dictionary
        self.sentiwords_txt = [line.replace('\n','').split(":") for
line in open("sentiwords_id.txt").read().splitlines()]
        self.sentiwords_dict = OrderedDict()
        for term in self.sentiwords_txt:
            self.sentiwords_dict[term[0]] = int(term[1])
        #create emoticon dictionary
        self.emoticon_txt = [line.replace('\n','').split(" | ") for
line in open("emoticon_id.txt").read().splitlines()]
        self.emoticon_dict = OrderedDict()
        for term in self.emoticon_txt:
            self.emoticon_dict[term[0]] = int(term[1])
        #create idioms dictionary
        self.idioms_txt = [line.replace('\n','').split(":") for line
in open("idioms_id.txt").read().splitlines()]
        self.idioms_dict = OrderedDict()
        for term in self.idioms_txt:
            self.idioms_dict[term[0]] = int(term[1])
        #create boosterwords dictionary
        self.boosterwords_txt = [line.replace('\n','').split(":") for
line in open("boosterwords_id.txt").read().splitlines()]
        self.boosterwords_dict = OrderedDict()
        for term in self.boosterwords_txt:
            self.boosterwords_dict[term[0]] = int(term[1])
        self.negation_conf = config["negation"]
        self.booster_conf = config["booster"]
        self.ungkapan_conf = config["ungkapan"]
        self.consecutive_conf = config["consecutive"]
        self.repeated_conf = config["repeated"]
        self.emoticon_conf = config["emoticon"]
        self.question_conf = config["question"]
        self.exclamation_conf = config["exclamation"]
        self.punctuation_conf = config["punctuation"]
        self.mean_conf = False

    def senti(self,term):
        try:
            return self.sentiwords_dict[term]
        except:
            return 0

    def emosikon(self,term):
        try:
```



```

        return self.emoticon_dict[term]
    except:
        return 0

def ungkapan(self,term):
    try:
        return self.idioms_dict[term]
    except:
        return 0

def booster(self, term):
    try:
        return self.boosterwords_dict[term]
    except:
        return 0

def cek_negationword(self, prev_term, prev_term2):
    #jika kata sebelumnya (index-1) adalah kata negasi, negasikan
    nilai -nya
    if prev_term in self.negasi or prev_term2+" "+prev_term in
    self.negasi:
        # print prev_term
        self.score = -abs(self.score) if self.score>0 else
    abs(self.score)

def cek_boosterword(self,term):
    booster_score = self.booster(term)
    if booster_score !=0 and self.score>0: self.score +=
    booster_score
    if booster_score !=0 and self.score<0: self.score -=
    booster_score

def cek_consecutive_term(self, prev_term):
    if self.prev_score>0 and self.score >=3: self.score+=1
    if self.prev_score<0 and self.score <=-3: self.score-=1

def cek_ungkapan(self, bigram,trigram, i):
    bigram = ' '.join(bigram)
    trigram = ' '.join(trigram)
    ungkapan_score = self.ungkapan(bigram)
    if ungkapan_score==0:
        ungkapan_score = self.ungkapan(trigram)
    if ungkapan_score!=0:
        self.score = ungkapan_score
        self.prev_score = 0
        self.pre_max_pos[i-1] = 1
        self.pre_max_neg[i-1] = -1
        self.max_pos = self.pre_max_pos[i-2] #if
    len(self.pre_max_pos)>1 else 1
        self.max_neg = self.pre_max_neg[i-2] #if
    len(self.pre_max_neg)>1 else -1
        self.sentence_score[i-1] = re.sub(r'\
    [\d\]', '',self.sentence_score[i-1])

def cek_repeated_punctuation(self, next_term):
    if re.search(r'!{2,}',next_term) and self.score >=3:
    self.score+=1

```

```

        if re.search(r'!\{2,}',next_term) and self.score <=-3:
self.score=-1

def remove_extra_repeated_char(self, term):
    return re.sub(r'([A-Za-z])\1{2,}',r'\1',term)
def plural_to_singular(self, term):
    return re.sub(r'([A-Za-z])+\1', r'\1',term)
def classify(self):
    result = "neutral"
    try:
        if self.mean_conf:
            mean_p = np.mean(self.mean_pos)
            mean_n = np.mean(self.mean_neg)
            print mean_p, mean_n
            if mean_p > mean_n:
                result = "positive"
            elif mean_p < mean_n and not self.is_tanya:
                result = "negative"
            elif mean_p < mean_n and self.is_tanya:
                result = "neutral"
        else:
            if abs(self.sentences_max_pos) >
abs(self.sentences_max_neg):
                result = "positive"
            elif abs(self.sentences_max_pos) <
abs(self.sentences_max_neg):
                result = "negative"
            elif abs(self.sentences_max_pos) ==
abs(self.sentences_max_neg):
                result = "neutral"
    except:
        print "error ",self.sentences_max_pos,
self.sentences_max_neg
        return result
    def cek_neutral_term(self,terms,i):
        if terms[i-1] in self.neutral_term or terms[i+1] in
self.neutral_term: self.score=1

def main(self,sentence):
    self.neutral_term = ['jika','kalau']
    sentences = sentence.split('.')
    self.sentences_max_neg = -1
    self.sentences_max_pos = 1
    self.sentences_score = []
    self.sentences_text = []
    for sentence in sentences:
        self.max_neg = -1
        self.max_pos = 1
        self.mean_neg = [1]
        self.mean_pos = [1]
        self.sentence_score=[]
        terms = sentence.split()
        # terms = re.split(r'[\s,.]',sentence)
        terms_length = len(terms)
        self.is_tanya = False
        self.sentence_text = ''
        # print self.max_pos, self.max_neg

```

```

minimal 2
#SEMUA KALIMAT YANG MEMILIKI TANDA SERU MEMILIKI +ve
if self.exclamation_conf and re.search('!',sentence):
self.max_pos = 2
self.prev_score = 0
self.pre_max_pos = []
self.pre_max_neg = []
for i,term in enumerate(terms):
# repeated_term = ''
is_extra_char = False
plural = ''
self.score = 0
# if re.search(r'[A-Za-z\-.]+',term):
# print term
if re.search(r'([A-Za-z])\1{3,}',term):
is_extra_char = True
# repeated_term =term
term = self.remove_extra_repeated_char(term)
if re.search(r'([A-Za-z]+)\-\1',term):
plural = term
term = self.plural_to_singular(term)
#GET SENTI SCORE#
self.score = self.senti(term)
# print "senti score",term, self.score

#NEGATION HANDLER#
if self.negation_conf and self.score !=0 and
i>0:self.cek_negationword(terms[i-1],terms[i-2])
# print "negation score",term, self.score

#BOOSTERWORD HANDLER#
if self.booster_conf and self.score !=0 and i>0 and
i<=(terms_length-1):self.cek_boosterword(terms[i-1])
if self.booster_conf and self.score !=0 and i>=0 and
i<(terms_length-1):self.cek_boosterword(terms[i+1])
# print "booster score",term, self.score

#IDIOM/UNGKAPAN HANDLER#
if self.ungkapan_conf and i>0 and i<=(terms_length-
1):self.cek_ungkapan([terms[i-1],term],[terms[i-2],terms[i-1],term],i)
# if self.ungkapan_conf and i>=0 and i<(terms_length-
1):self.cek_ungkapan([term,terms[i+1]])
# print "idiom score",term, self.score

#CONSECUTIVE SENTIMENT WORD#
if self.consecutive_conf and i>0 and i<=(terms_length-
1) and self.score !=0:self.cek_consecutive_term(terms[i-1])
# print "consecutive score",term, self.score

#+1 SENTI SCORE IF REPEATED CHAR ON POSITIVE/NEGATIVE
+2 IF NEUTRAL TERM
if self.repeated_conf and is_extra_char==True and
self.score>0: self.score+=1
if self.repeated_conf and is_extra_char==True and
self.score<0: self.score-=1
if self.repeated_conf and is_extra_char==True and
self.score==0: self.score=2

```

```

        # print "repeat char score", term, self.score
        if self.punctuation_conf and i>=0 and i<(terms_length-
1): self.cek_repeated_punctuation(terms[i+1])
        # CEK APAKAH TERDAPAT KATA TANYA
        if self.question_conf and (term in self.tanya or
re.search(r'\?',term)):self.is_tanya = True
        # CEK neutral term
        if self.score!=0 and i>1 and i<(terms_length-2):
self.cek_neutral_term(terms,i)
        # if self.score!=0 and i>0 and i<(terms_length-4):
self.cek_neutral_term(terms,i)
        if self.emoticon_conf and self.score==0: self.score =
self.emosikon(term)

        self.prev_score = self.score
        if self.mean_conf and self.score>0:
self.mean_pos.append(self.score)
        if self.mean_conf and self.score<0:
self.mean_neg.append(abs(self.score))
        #GET MAX SCORE +ve/-ve
        self.max_pos= self.score if self.score > self.max_pos
else self.max_pos
        self.max_neg= self.score if self.score < self.max_neg
else self.max_neg
        #insert score info current term
        self.pre_max_pos.append(self.max_pos)
        self.pre_max_neg.append(self.max_neg)
        # print self.pre_max_pos, self.pre_max_neg
        if plural !='': term = plural
        self.sentence_text += ' {}'.format(term)
        if self.score != 0:term = "{} [{}]" .format(term,
self.score)
        self.sentence_score.append(term)

        self.sentences_text.append(self.sentence_text)
        self.sentences_score.append(" ".join(self.sentence_score))
        if self.is_tanya:
            self.max_neg = -1
            self.sentences_max_pos = self.max_pos if self.max_pos >
self.sentences_max_pos else self.sentences_max_pos
            self.sentences_max_neg = self.max_neg if self.max_neg <
self.sentences_max_neg else self.sentences_max_neg
            # print self.sentences_max_pos, self.sentences_max_neg
            sentence_result = self.classify()
            # print self.sentences_text
            return {"classified_text":".
".join(self.sentences_score),"tweet_text":".
".join(self.sentences_text),"sentence_score":self.sentences_score,"max
_positive":self.sentences_max_pos,"max_negative":self.sentences_max_ne
g,"kelas":sentence_result}

config = dict()
config["negation"] = True
config["booster"] = True
config["ungkapan"] = True
config["consecutive"] = True
config["repeated"] = True

```

```
config["emoticon"] = True
config["question"] = True
config["exclamation"] = True
config["punctuation"] = True
senti = sentistrength(config)
print senti.main("agnezmo pintar dan cantik sekali tetapi lintah darat
:)")
```