

1 Main-Algorithm

Data:

WV_0 : bestehende, normalisierte Liste von Wortvektoren

$Tweets$: Liste von Tweets

$Stopwords$: Liste von Stoppwörtern

Result:

WV_1 : verbesserte, normalisierte Liste von Wortvektoren

```
1  $WV_1 \leftarrow \emptyset$ ;  
2 foreach  $Tweet$  in  $Tweets$  do  
3    $V_{Tweet} \leftarrow (0, 0, 0, 0, 0, 0, 0)$ ;  
4   forall the  $Token$  in  $Tweet$  do  
5     if  $Token \notin Stopwords$  then  
6        $V_{Tweet} \leftarrow V_{Tweet} + WV_0(Token)$ ;  
7     end  
8   end  
9   forall the  $Token$  in  $Tweet$  do  
10    if  $Token \notin Stopwords$  then  
11       $WV_1(Token) \leftarrow WV_1(Token) + V_{Tweet}$ ;  
12    end  
13  end  
14 end  
15 foreach  $V_{Word}$  in  $WV_1$  do  
16    $V_{Word} \leftarrow \text{normalize}(V_{Word})$ ;  
17 end  
18 if  $\text{difference}(WV_0, WV_1) < \text{significance level}$  then  
19    $WV_0 \leftarrow WV_1$ ;  
20   Goto Line 2;  
21 else  
22   return  $WV_1$ ;  
23 end
```

Algorithm 1: Tweeegion Main-Algorithm

2 Geo-Algorithm

Data:

Tweets: Liste von Tweets mit Geo-Informationen

Stopwords: Liste von Stoppwörtern

Result:

WV: normalisierte Liste von Tokenvektoren

```
1  $WV \leftarrow \emptyset$ ;  
2 foreach Tweet in Tweets do  
3    $n \leftarrow \text{classify}(\textit{Tweet})$ ;  
4    $V_{\textit{Tweet}} \leftarrow \text{createVector}(n)$ ;  
5   forall the Token in Tweet do  
6     if Token  $\notin$  Stopwords then  
7        $WV(\textit{Token}) \leftarrow WV(\textit{Token}) + V_{\textit{Tweet}}$ ;  
8     end  
9   end  
10 end  
11 foreach  $V_{\textit{Word}}$  in WV do  
12    $V_{\textit{Word}} \leftarrow \text{normalize}(V_{\textit{Word}})$ ;  
13 end  
14 return  $WV_1$ ;
```

Algorithm 2: Tweegion Geo-Algorithm