

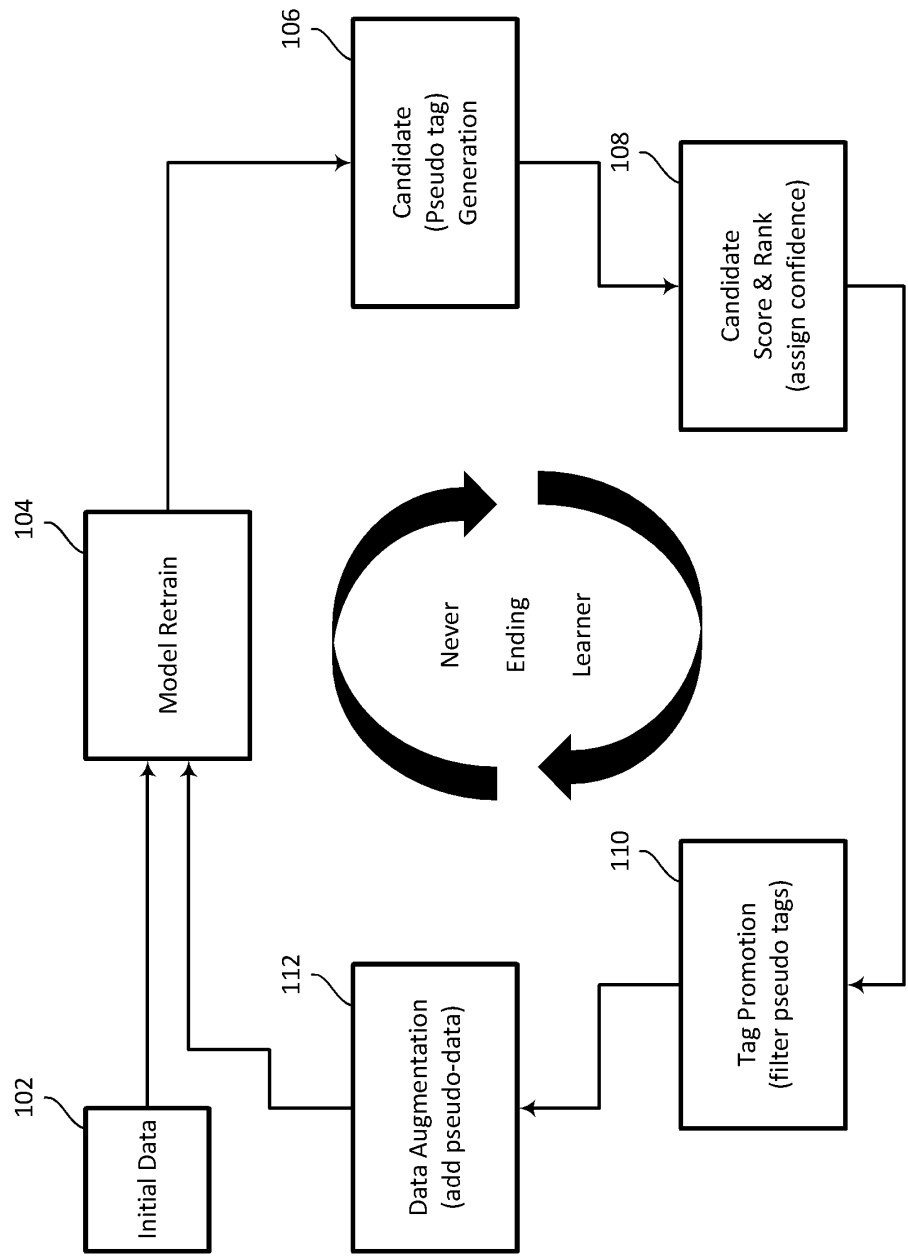


FIG. 1

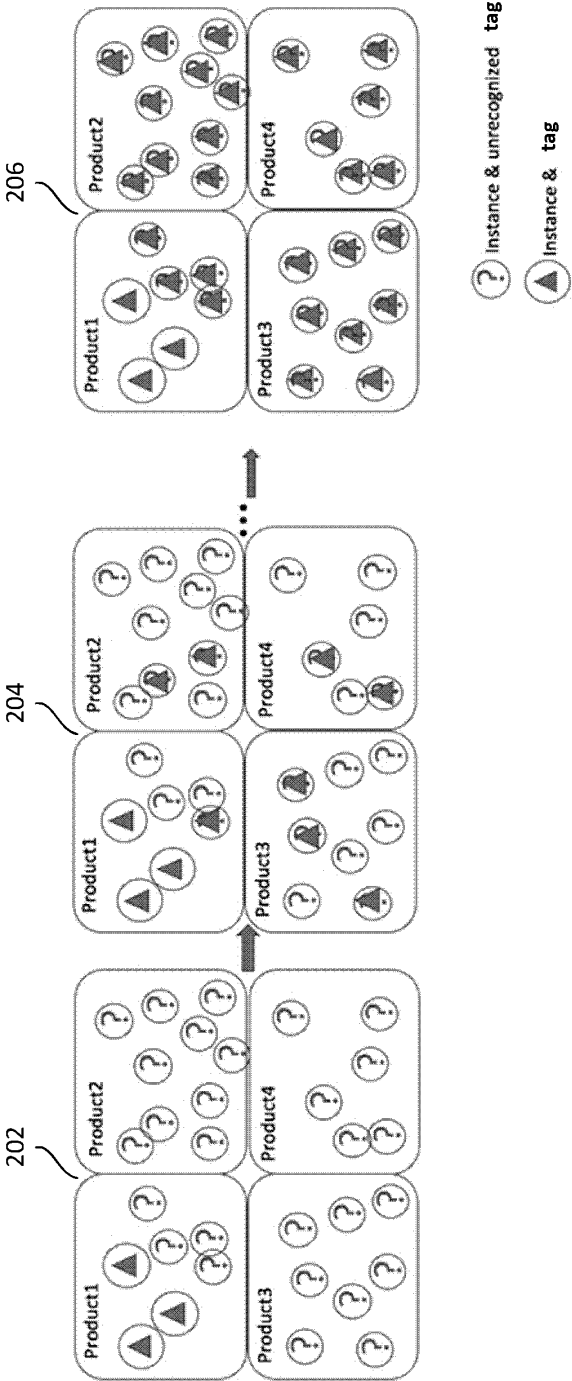


FIG. 2

300

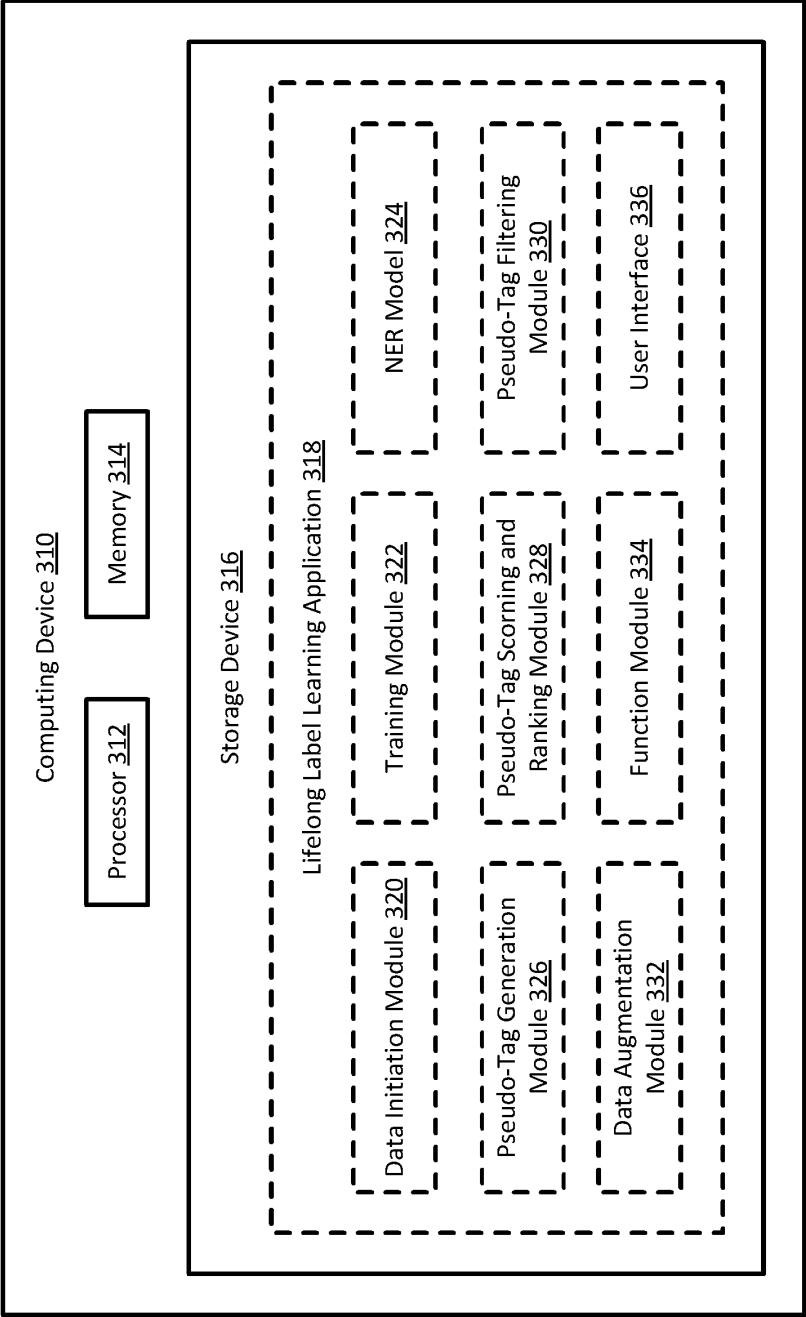


FIG. 3

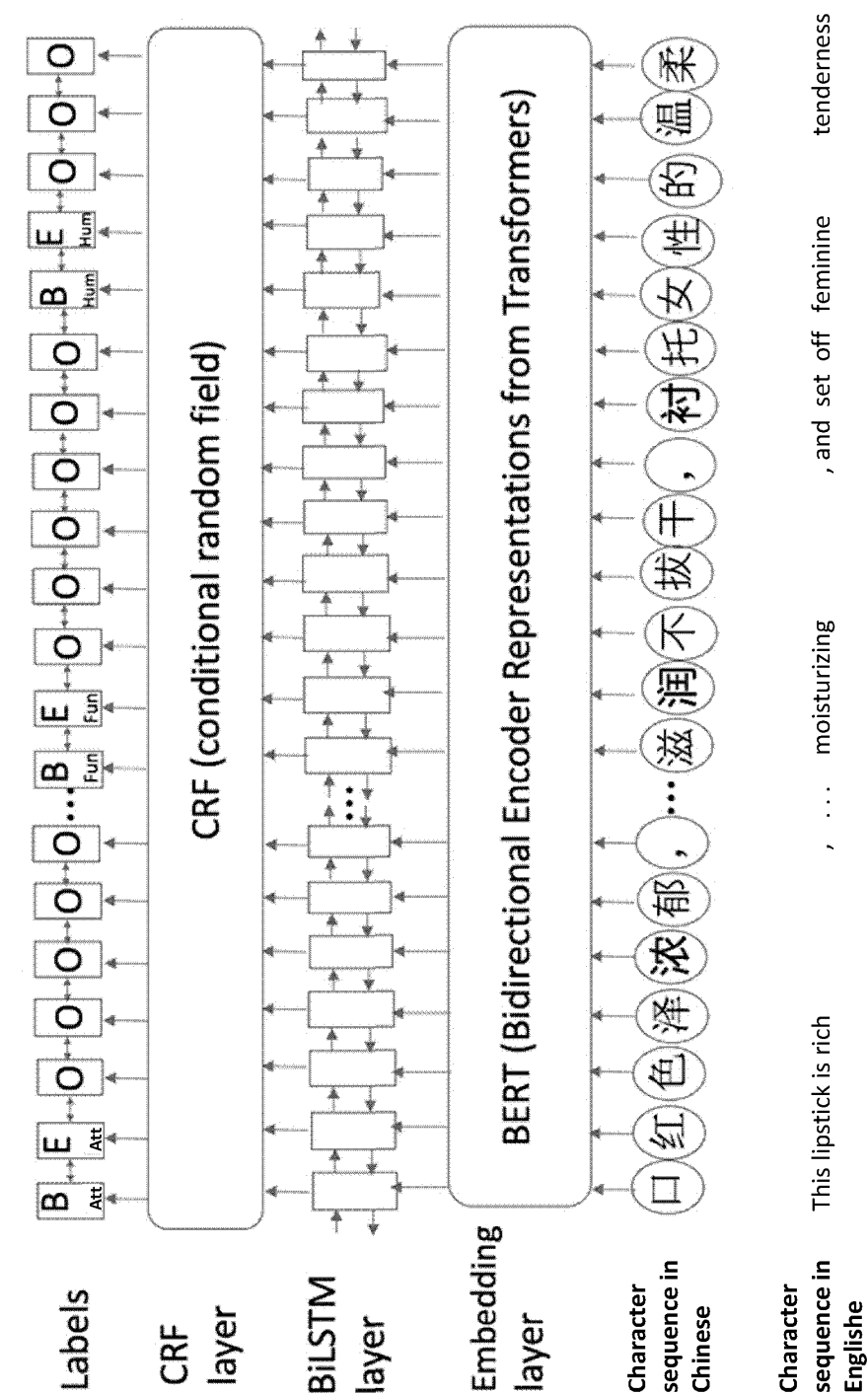


FIG. 4

500A

Household Appliance > Large Appliance >									
Brand	SIEMENS		HAIER		ZF		Mi		
	VINOCAVE		LITTLESWAN		EKEA		SKYWORTH		
Screen size	78 Inches & Up		70-75 Inches		65 Inches		58-60 Inches		55 In
Washer Capacity	20 Kg & Up		10-20 Kg		9.1-9.9 Kg		9 Kg		8.1-8.9 Kg
Frequency Type	Constant		Variable						
Advanced	Heating & Cooling Type >		Motor Type >		Height >		Energy Efficiency Rating >		Product Feature >
	Automatic Self-Cleaning		Quiet Air Conditioner		Cylindrical Air Conditioner		Independent Dehumidification		Easy to Take part and Wash

FIG. 5A

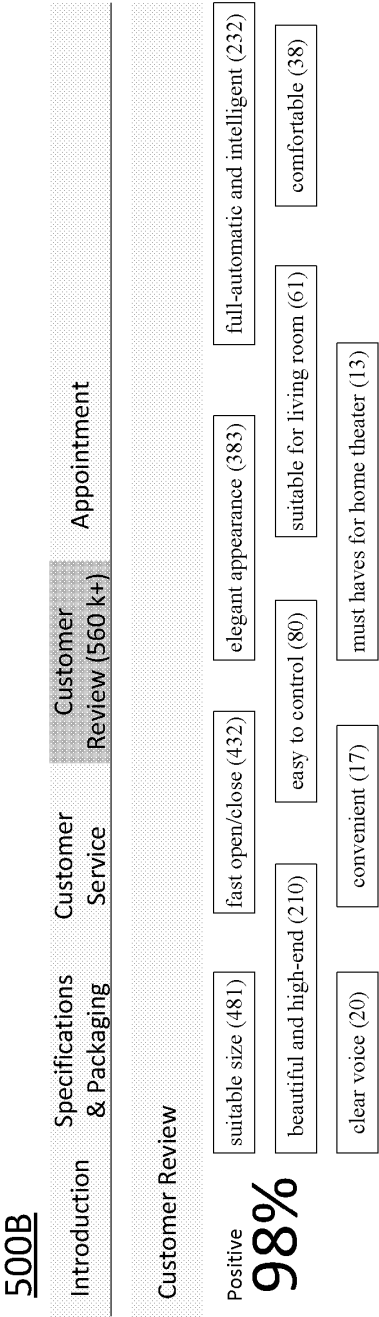


FIG. 5B



FIG. 5C

7 / 9

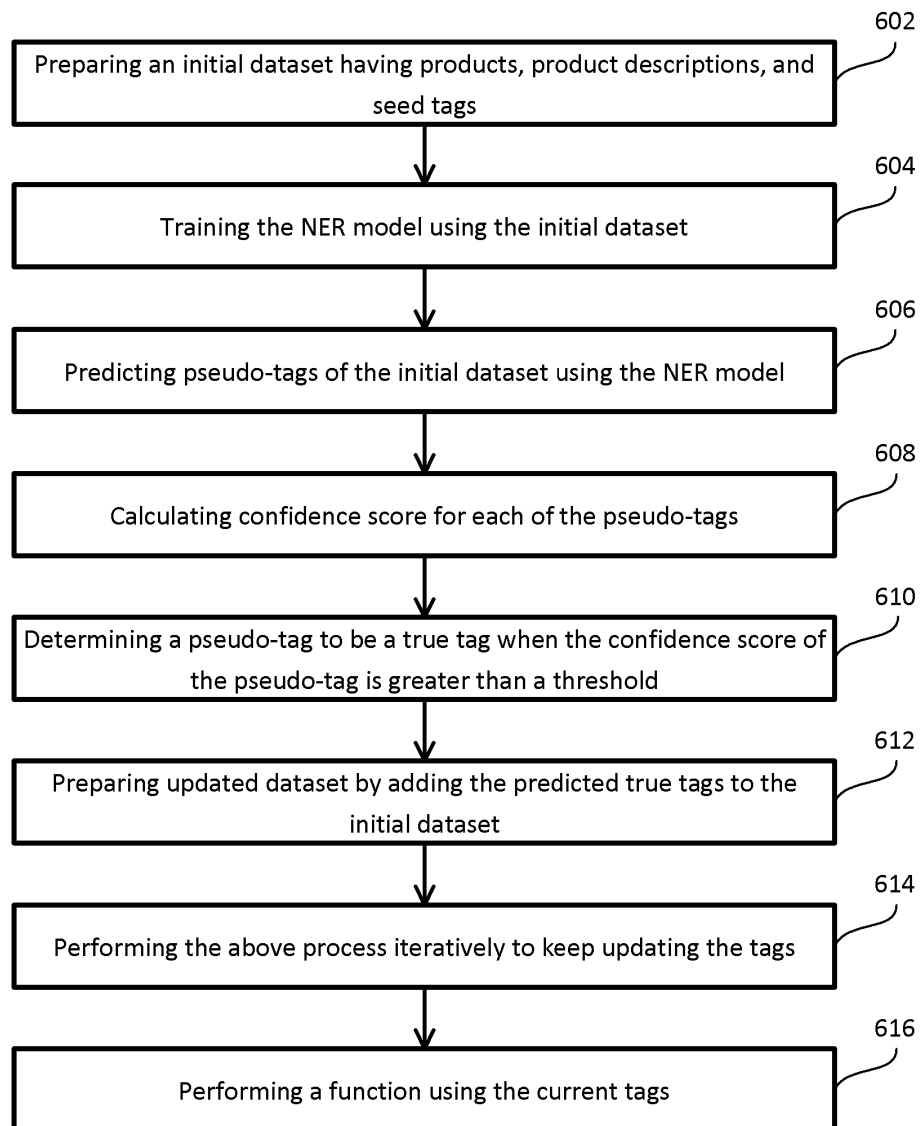
600

FIG. 6

8 / 9

```

1  Input:
2  unstructured_text_old_prds /* a set of unstructured text for each old product */,
3  unstructured_text_new_prds /* a set of unstructured text for each new product */,
4  list_of_old_prds /* a list of product categories */,
5  list_of_new_prds /* a list of product categories */,
6  bootstrap_tags_of_old_prds /* a set of bootstrap tags as seed tags for each old product */
7  Output: a set of tags for each product
8  Procedure:
9  Begin
10 /* tbd_list_of_categories is to do list of product categories to learn tags */
11 tbd_list_of_categories = list_of_old_prds + list_of_new_prds
12 /* unstructured_text is all unstructured text for all corresponding product categories */
13 unstructured_text = unstructured_text_old_prds + unstructured_text_new_prds
14
15 While True: /* never-ending tag learning */
16   If len(tbd_list_of_categories) == 0:
17     break /* if to do list is null, it means all product categories have enough good tags, so stop learn */
18   /* use bootstrap tags and all unstructured text to generate dataset */
19   dataset = generate_dataset(bootstrap_tags_of_old_prds, unstructured_text);
20   model = train_NER_model(dataset) /* train the NER model using the new generated dataset */
21
22   for cate in tbd_list_of_categories: /* iterate tbd_list_of_categories of product categories */
23     /* use the model to predict the entities in the inputted unstructured text as pseudo tags */
24     pseudo_tags = model.predict(unstructured_text)
25     for p_tag in pseudo_tags: /* iterate all pseudo tags */
26       /* calculate the confidence score for a pseudo tag */
27       conf_score_of_p_tag = cal_conf(p_tag, unstructured_text)
28       /* get bootstrap tags for this product category, if it is new product category, bootstrap_tags is null */
29       bootstrap_tags = get_tags(bootstrap_tags_of_old_prds)
30       if conf_score_of_p_tag > threshold_of_tag_promotion:
31         /* if the confidence score of a pseudo tag is bigger than the promotion threshold */
32         bootstrap_tags.append(p_tag) /* add pseudo tag into bootstrap tags of this category */
33     end for /* end of for-loop for all pseudo tags */
34
35     if condition_of_stop_learning(bootstrap_tags):
36       /* if meet the condition of stopping learning tag */
37       /* delete this product from to_do_list of product categories */
38     else:
39       if cate in list_of_new_prds:
40         list_of_old_prds.append(cate) /* if it is new product, add it into old product list */
41     end for /* end of for-loop for all product categories */
42   /* continue with next iteration tag learning to increase recall for this category in to-do-list product categories */
43 end while /* end of never-ending tag learning */
44
45 End
46
47

```

FIG. 7

9 / 9

# of tags Products	V0	V1	V2	V3	V4	V5
Garment & Large appliances	1011	47	76	74	85	65

FIG. 8

# of tags Products	V0	V1
Emulsion facial cream	0	146
Makeup remover	0	93
Cleanser	0	109
Toner	0	143
Eyebrow pencil	0	75
Eyecream	0	116
Essence	0	140
Manicure	0	68
Sunscreen	0	110
Mask sheet	0	113
perfume	0	104
Desk	0	95
Bed	0	127
Sofa	0	126
TV stands	0	99
Portable closet	0	41
Wardrobe	0	108
Show Storage Cabinet	0	109
Dining Table	0	122

FIG. 9