SemEval 2015 Task 12
Aspect Based Sentiment Analysis (ABSA) Evaluation & Baselines

ABSA 2015 task

Given a review text about a laptop or a restaurant the goal in ABSA 2015 is to identify tuples that contain the following types of information[1]:

- **Slot 1**: Aspect Category (Entity and Attribute). Identify every entity E and attribute A pair E#A towards which an opinion is expressed in the given text. Each E#A pair defines an aspect category of the given text.
- **Slot 2 (Only for the restaurants domain)**: Opinion Target Expression (OTE). An opinion target expression (OTE) is an expression used in the given text to refer to the reviewed entity E of a pair E#A.
- **Slot 3:** Sentiment Polarity. Each identified E#A category in laptops or <category, target> pair in restaurants has to be assigned a polarity, from a set P = {positive, negative, neutral}.

Examples of opinion tuples for the restaurant domain are shown in Fig 1.

Review id:"1004293"

Judging from previous posts this used to be a good place, but not any longer.
{category:"RESTAURANT#GENERAL", target:"NULL", from:"-", to="-", polarity:"negative" }

We, there were four of us, arrived at noon - the place was empty - and the staff acted like we were imposing on them and they were very rude.
{category:"SERVICE#GENERAL", target:"staff", from:"75", to:"80", polarity:"negative" }

They never brought us complimentary noodles, ignored repeated requests for sugar, and threw our dishes on the table.
{category:"SERVICE#GENERAL", target:"NULL", from:"-", to:"-", polarity:"negative"}

The food was lousy - too sweet or too salty and the portions tiny.
{category="FOOD#QUALITY", target:"food", from:"4", to:"8", polarity="negative" }
{category:"FOOD#STYLE_OPTIONS", target:"portions", from:"52", to:"60", polarity:"negative" }

After all that, they complained to me about the small tip.
{category:"SERVICE#GENERAL", target:"NULL", from:"-", to:"-", polarity:"negative" }

Avoid this place!
{category:"RESTAURANT#GENERAL", target:"place", from:"11", to:"16", polarity:"negative"}

Figure 1: ASBA 2015 opinion tuples for a restaurant review.

[1] http://alt.qcri.org/semeval2015/task12/

Evaluation

Slot 1: The evaluation assesses whether a system identifies and returns the set of aspect categories towards which an opinion is expressed. In particular, precision, recall and F-1 scores are calculated by comparing the list of the categories that a system returned (for a sentence) to the corresponding gold list. These lists are constructed by extracting the values of Slot 1 (category). For example for the 4th sentence of Fig 1 the list is {(FOOD#QUALITY), (FOOD#STYLE_OPTIONS)}. The calculation ignores duplicate occurrences of categories. For example, for the following sentence the categories list is {(FOOD#QUALITY)}.

Furthermore, the rice had no seasoning, so the sushi was bland and disgusting. {category="FOOD#QUALITY", target="rice, from="17", to="21", polarity="negative"} {category="FOOD#QUALITY", target="sushi", from="47", to="52", polarity="negative"}

You can evaluate your system in category extraction by running the following command[2].

java -cp ./A.jar absa15.Do Eval ./pred.xml ./teGld.xml 1 0

pred.xml contains the predicted annotations and teGld.xml the gold annotations. Both xml files should be in the same format as the provided training data[3].

Slot 2: The evaluation assesses whether a system identifies and returns the set of targets, i.e. the expressions that are used in a sentence to refer to the reviewed entities. In particular, precision, recall and F-1 scores are calculated by comparing the list of the targets that a system returned (for a sentence) to the corresponding gold list. These lists are constructed using the target offsets. For example for the 4th sentence of Fig 1 the extracted list is {(4, 8), (52, 60)}. The calculation discards NULL targets since they do not correspond to explicit target mentions. For example, for the following sentence the constructed list is {(19,29)}.

Terrible, terrible management - deserves to be shut-down. {category="SERVICE#GENERAL", target="management", from="19" to="29", polarity="negative"} {category="RESTAURANT#GENERAL", target="NULL", from="-", to="-", polarity="negative"}

[2] A.jar is included in the package that is provided with this document.
[3] http://alt.qcri.org/semeval2015/task12/index.php?id=data-and-tools

Duplicate targets are also ignored[4], so for the next sentence the target list is {(51, 55)}

I expected quite a bit more from such an expensive menu. {category="FOOD#PRICES ", target=" menu ", from="51", to="55", polarity=" negative"} {category="FOOD#QUALITY", target=" menu ", from="51", to="55", polarity=" negative"}

You can evaluate your system in target extraction by running the following command.

java -cp ./A.jar absa15.Do Eval ./pred.xml ./teGld.xml 2 0

Slot 1&2: <category, target> evaluation assesses whether a system identifies the targets, the aspects categories and constructs the corresponding tuples. Again precision, recall and F1 scores are calculated by comparing the <category, target> tuples of a system to the gold ones. You can evaluate your system in <category, target> extraction by running the following command.

java -cp ./A.jar absa15.Do Eval ./pred.xml ./teGld.xml 3 0

The compared lists in this case contain the target offsets and the category values. For example for the 4th sentence this list is the following.

{(FOOD#QUALITY,4,8), (FOOD#STYLE_OPTIONS,52,60)}.

Slot 3: For polarity classification evaluation we use the total accuracy score. To evaluate your system run the command shown below. The pred.xml should contain the gold annotations for the categories and targets and the corresponding predicted polarities.

java -cp ./A.jar absa15.Do Eval ./pred.xml ./teGld.xml 5 1

The program will print the total accuracy score as well as precision, recall and F1 scores for each polarity label (positive, negative, neutral).

Validation

To check whether an xml file generated by a system (e.g. pred.xml) is well formed and all the slots are filled with valid values you can run the command that is shown below. The first argument is the xml to be checked (pred.xml), the second is an xsd file (ABSA15.xsd) and the third the domain (lapt|rest).

java -cp ./A.jar absa15.Do Validate ./pred.xml ./ABSA15.xsd lapt

[4] A target is defined by its starting and ending offset.

The script validates the xml against the xsd and checks the slot values. For example, if slot 1 (category) is filled with a value that does not correspond to the E,A inventories[5] of the domain a relevant message will be printed. Similarly, if slot 3 (polarity) is assigned a value not belonging to the set P = {positive, negative, neutral} a relevant message will also be printed.

Baselines Description

We have implemented 3 baselines for the respective slots of the tuples that are required in ABSA 2015.

Slot 1: Aspect Category (Entity and Attribute). For category extraction a Support Vector Machine (SVM) model with linear kernel is learnt. In particular, *n* bag-of-words (BOW) features[6] are extracted from the respective sentence of each <category, target, polarity> tuple that is encountered in the training data. As a label for the feature vector the category value (e.g. SERVICE#GENERAL) of the tuple is used. Similarly, for each test sentence *s* one BOW feature vector is built and the trained SVM model is used to predict the probabilities of assigning each possible category to *s* (e.g. {SERVICE#GENERAL, 0.2}, {RESTAURANT#GENERAL, 0.4},…,{FOOD#STYLE_OPTIONS, 0.4}). Then, a threshold t is used to decide which of them will be assigned to *s*[7].

Slot 2: Opinion Target Expression (OTE). This baseline uses the training reviews to create for each category *c* (e.g. SERVICE#GENERAL) a list of targets to which it is linked to. For example, SERVICE#GENERAL is linked to "staff" in the examples of Fig 1. Then, given a test sentence *s* and a category *c*, the baseline finds the first occurrence in *s* of each target encountered in *c*'s list. Finally, the target slot of *c* is filled with the first target occurrence in *s*. If no target occurrences are found, the slot is assigned the value NULL.

Slot 3: Sentiment Polarity. For polarity prediction we train an SVM classifier with linear kernel. Again, as in Slot 1, *n* BOW features are extracted from the respective sentence of each <category, target, polarity> tuple of the training data. In addition, a feature that indicates the category of the tuple is used (*n*+1 features in total). As a label for the extracted feature vector the corresponding polarity value is used (positive|negative|neutral). Then, for each opinion tuple (category, target, -) of a test sentence *s* a feature vector is built and it is classified using the learnt SVM model.

[5] For more information on E,A inventories see the guidelines that are provided in http://alt.qcri.org/semeval2015/task12/index.php?id=data-and-tools

[6] the n most frequent tokens of the training set are used.

[7] we use the –b 1 option of LibSVM package to obtain probability estimates.

ABSA 2015 baselines and evaluation package

The baseline systems and evaluation scripts are available for download as a single zip (BaseEvalValid.zip) from SemEval-2015 Task 12 website[8]. They are implemented in java and integrated into a Linux shell script (absa15.sh). The script uses the LibSVM software[9] for SVM training and prediction.

Installation and running: To install the package extract BaseEvalValid.zip to a directory (e.g. BaseEvalValid). Then, open a terminal and move to the top level directory (e.g. BaseEvalValid) of the package. Move to libsvm-3.18 directory by typing "cd libsvm-3.18" and run "make" to build the "svm-train" and "svm-predict" programs. After this, return to the top level directory by typing "cd ..". Before running absa15.sh (./bash15.sh) you have to open absa15.conf with a text editor and set the required parameters as it is described below.

src It should be assigned the name of the xml file that contains the training data of a domain (laptops or restaurants). The **src** xml file should be placed to the top level directory of the package (the one that contains the absa15.sh).

e.g. src=ABSA-15_Restaurants_Train_Final.xml

src=ABSA-15_Laptops_Train_Data.xml

dom: It should be assigned the domain that the **src** file corresponds to.

e.g. dom=rest for restaurants and dom=lapt for laptops.

thr: It sets the threshold for assigning categories in a sentence.

e.g. thr=0.12

sfl: It indicates whether the reviews of the **src** file will be shuffled or not before splitting them into parts and generating the corresponding files. **slf** should be set to 0 for not shuffling or to 1 for shuffling. The package uses a constant seed so the same review order is generated in every run.

xva: It indicates whether a cross validation will be performed on the input data (**src** file). For running cross validation **xva** should be set to 1. For just splitting the **src** file in a training and test part it should be set to 0.

fld: It specifies the number of chunks that the reviews of the **src** file will be split into. These generated chunks contain whole reviews and have approximately the same number of opinion tuples. If **xva** is set to 0 then one of chunks will be used for testing and the rest fld-1 for training. The chunk that will be used for testing is specified by the **partIdx** parameter that is described below. Otherwise, if **xva** is set to 1 the script will run fld rounds (iterations), in each round one of generated chunks is used for testing and the other fld-1 for training.

[8] http://alt.qcri.org/semeval2015/task12/index.php?id=data-and-tools
[9] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

partIdx: When **xva** is set to 0 it partIdx specifies the chunk index that will be used for testing. For example, if fld value is 10 then it can take a value from 0 to 9.

ftr: It specifies the number of BOW features that will be used to create the SVM training and testing vectors.

ttd: It specifies the name of the folder where the SVM training and testing files will be written. In this folder they also are also stored the input xml files as well the outputs of the baselines. This folder should be placed in the top level directory of the package.

Inputs and outputs: When **xva** parameter is set to 0 the script operates as follows:

The src file reviews are split and the files that are listed below are generated. The contents of these files are determined from the values of **fld** and **partIdx** parameters**.**

- tr.xml: It contains the training reviews with the human annotations.
- teCln.xml: It contains the testing reviews without any human annotations.
- teCln.PrdAspTrg.xml: It contains the testing reviews along with the gold category and target annotations.
- teGld.xml. It contains the testing reviews and all the corresponding gold annotations (category, target, polarity).

Then, in a phase A the script predicts the category (Slot 1) and target[10] (Slot 2) annotations for the sentences of teCln.xml and stores the result to **teCln.PrdAspTrg.xml**. Subsequently, in a phase B it predicts the polarity of each gold (category, target) tuple that is stored in **teGldAspTrg.xml** and generates **teGldAspTrg.PrdPol.xml[11].** Finally, the outputs (**teCln.PrdAspTrg.xml, teGldAspTrg.PrdPol.xml**) are compared to the gold annotations of **teGld.xml** using the evaluation measures/scripts that were described in the previous section.

Evaluation scores for baselines systems

For both domains we have set **sfl**=1, **xva**=0, **fld**=10, **partIdx**=9 in absa15.conf and run asba15.sh. We have set to 1000 the number of BOW features to be used in SVM training and prediction (ftr=1000). The script shuffles the reviews and splits the result into 10 chucks (parts). It uses the 10th part for testing and the remaining nine for training. The results we obtain for slots 1 and 2 if we set the threshold for category prediction to 0.2 for the restaurants and to 0.12 for the laptops are the following:

[10] Only for the restaurants domain.

[11] Similar inputs and outputs are produced in cross validation mode, for example, for round 1 the script generates tr.xml.0, teGld.xml.0, teGldAspTrg.xml.0 etc.

Domain	Evaluated slot(s)	Scores
Restaurants	category	PRE=0.5341615 REC=0.688 **F-1 =0.60139865**
Restaurants	target	PRE=0.55421686 REC=0.43396226 **F-1=0.48677248**
Restaurants	(category, target)	PRE=0.36024845 REC=0.42028984 **F-1=0.38795984**
Laptops	category	PRE=0.35858586 REC=0.41764706 **F-1 =0.38586956**

The polarity classification results for the gold (category, target) tuples are the following:

Domain	Evaluated slot(s)	Scores
Restaurants	polarity	**Accuracy= 0.7173913**
Laptops	polarity	**Accuracy= 0.7647059**