Labels



Identifying poorly performing keywords can be a tedious task. This script identifies poorly performing keywords and label them for later review. Additionally, the script will apply a different label for "brand" versus "non-brand" keywords and send an email summary each time new keywords have been added.

How it works

**Finding poorly performing keywords**

The script uses a helper function that runs the same algorithm against your keywords with different performance criteria:

var lowBrandPerformers = labelLowPerformers(
    'brand-review',
    // 2% ctr
    0.02,
    // $1 (or account currency)
    1,
    // Only look at branded keywords
    'LabelNames CONTAINS\_ANY ["brand-keyword"]');

We want to mark branded keywords for review if they have less than 2% CTR and average CPC of more than $1 (note that this will use your account's currency). The LabelNames condition allows us to only look at branded keywords. Modify these values to best suit your account and advertising goals.

We use a selector that not only identifies keywords that have a CTR below our threshold, but also only those keywords that are eligible to trigger impressions:

var keywords = AdWordsApp.keywords().
    // Only want to look at keywords eligible to trigger ads.
    withCondition('CampaignStatus = ENABLED').
    withCondition('AdGroupStatus = ENABLED').
    withCondition('Status = ENABLED').
    // Filter out keywords above performance threadhold
    withCondition('Ctr < ' + minCtr).
    withCondition('AverageCpc > ' + maxCpc).
    withCondition(labelCondition).
    forDateRange('LAST\_7\_DAYS').
    get();

Next, we apply the new label. If applyLabel('foo') is invoked on an object that already has this label, no operation will be run nor error generated. However, here we're checking if the label already exists so we know if the keyword was already designated for review.

var keyword = keywords.next();
// We should mark it for review depending on whether it's a branded keyword or not.
var label = isBrand(keyword.getText()) ? 'brand-review' : 'non-brand-review';
// Check if it already has the label.
if (!hasLabel(keyword, label)) {
  keyword.applyLabel(label);
  // Track it as newly labeled.
  newlyLabeled.push(keyword);
}

After we've gone through all our keywords, the newlyLabeled array will have references to all the keywords we’ve recently applied labels to. This allows us to send an email summary indicating any new keywords to review.

// Send an email notifying that there are new keywords to review.
if (EMAIL\_ADDRESS && newlyLabeled.length) {
  MailApp.sendEmail(EMAIL\_ADDRESS, "New keywords to review", generateEmail(newlyLabeled));
}

**Working with labels**

Labels must be created via [AdWordsApp.createLabel](https://developers.google.com/adwords/scripts/docs/reference/adwordsapp/adwordsapp#createLabel_3) before you apply them to keywords, campaigns, etc. If you try to apply a label before it has been created, you'll receive an error (that particular operation will fail, but the script will continue). You will also receive an error if you try to create a label that already exists—but the script will continue and you’ll be able to apply that label to entities.

This code will create three new keywords on the first run, but will see error messages (due to the labels having already been created) if run more than once.

AdWordsApp.createLabel('brand-review', 'Branded keywords needing review', 'red');
AdWordsApp.createLabel('non-brand-review', 'Branded keywords needing review', 'maroon');
AdWordsApp.createLabel('brand-keyword', 'Keywords that are part of our brand', 'blue');

The function that checks for labels on a given keyword uses a selector to filter on LabelName:

function hasLabel(keyword, label) {
 return keyword.labels().withCondition("Name = '" + label + "'").get().hasNext();
}

**Branding**

Handling branded keywords differently from non-branded keywords helps focus analysis. This script uses a simple check to see if the keyword has one of a number of phrases.

function isBrand(s) {
  if (!s) {
    return false;
  }
  for (var i = 0; i < BRAND\_NAMES.length; i++) {
    if (s.toLowerCase().indexOf(BRAND\_NAMES[i].toLowerCase()) != -1) {
      return true;
    }
  }
  return false;
}

The labelBrandedKeywords function uses the isBrand function to label all branded keywords with the appropriate label:

function labelBrandedKeywords() {
  var keywords = AdWordsApp.keywords().get();
  while (keywords.hasNext()) {
    var keyword = keywords.next();
    if (isBrand(keyword.getText())) {
      keyword.applyLabel('brand-keyword');
    }
  }
}

**Previewing and email summary**

After creating the new labels and applying labels to branded keywords, you can now run the main script logic in checkPerformance to mark keywords for review:

function main() {
  //createLabels();

  //labelBrandedKeywords();

  checkPerformance();
}

You will also receive an email listing all the keywords that have been marked for review by the script. Since this script labels your keywords, you can later log in to the AdWords UI to see if you want to remove or adjust the bidding on your keywords. You can pause the keyword so the script will ignore it on future executions.

Scheduling

Scheduling this script to run daily or weekly will let you know when you have poorly performing keywords to review.

Setup

* Create a new AdWords script with the source code below.
* Modify the values at the beginning of the script to your purpose. For example, if you owned a fudge shop named Elmer's Fudge, you could set the BRAND\_NAMES variable as follows so the different variants of your brand are accounted for:

var BRAND\_NAMES = ['Elmer s Fudge', 'Elmer Fudge', 'Elmer s Fudge shop', 'Elmer' ];

You should also change the EMAIL\_ADDRESS to your own to get a summary email:

var EMAIL\_ADDRESS = 'your.email@example.com';

* Before previewing your script, you'll need to initialize your labels. Change the main function to look like this:

function main() {
  createLabels();

  //labelBrandedKeywords();

  //checkPerformance();
}

* Run the script. This will create the labels that will be applied to keywords.
* Label your branded keywords. Change the main function to this:

function main() {
  //createLabels();

  labelBrandedKeywords();

  //checkPerformance();
}

* Run the script—this will now apply labels to all keywords that match your criteria for branding. You can also click **Preview** to see which keywords will get labeled as brand keywords.

[Creating an AdWords script](https://developers.google.com/adwords/scripts/docs/solutions/labels)

Source code

// List of phrases that indicate a keyword is part of your brand.
// In this example, we operate a fudge store named Elmer Fudge.
// Keywords with our brand name are "branded" keywords, those without "non-brand".
var BRAND\_NAMES = ['Elmer Fudge', 'Elmer'];
// Set this to a your email address to get an email summary of new keywords to review.
var EMAIL\_ADDRESS = 'email@example.com';

function main() {
  //createLabels();

  //labelBrandedKeywords();

  //checkPerformance();
}

function checkPerformance() {
  var lowBrandPerformers = labelLowPerformers(
      'brand-review',
      // 2% ctr
      0.02,
      // $1 (or account currency)
      1,
      // Only look at branded keywords
      'LabelNames CONTAINS\_ANY ["brand-keyword"]');
  var lowNonBrandPerformers = labelLowPerformers(
      'brand-review',
      // 1% ctr
      0.01,
      // $2 (or account currency)
      2,
      // Only look non-branded keywords
      'LabelNames CONTAINS\_NONE ["brand-keyword"]');

    // Send an email notifying that there are new keywords to review.
  if (EMAIL\_ADDRESS && (lowBrandPerformers.length || lowNonBrandPerformers.length)) {
    var body = 'Branded keywords\n' + generateEmail(lowBrandPerformers) +
        'Non-Branded keywords\n' + generateEmail(lowNonBrandPerformers);
    MailApp.sendEmail(EMAIL\_ADDRESS, "New keywords to review", body);
  }
}

/\*\*
 \* Labels keywords that have a low CTR and high CPC as a poor performer.
 \*
 \* @param {string} labelName Label to apply to poor performers.
 \* @param {number} minCtr Minimum CTR we expect keywords to have.
 \* @param {number} maxCpc Maximum average CPC.
 \*/
function labelLowPerformers(labelName, minCtr, maxCpc, labelCondition) {
  // Keep track of keywords that have had a label added this run.
  var newlyLabeled = [];
  var keywords = AdWordsApp.keywords().
      // Only want to look at keywords eligible to trigger ads.
      withCondition('CampaignStatus = ENABLED').
      withCondition('AdGroupStatus = ENABLED').
      withCondition('Status = ENABLED').
      // Filter out keywords above performance threadhold
      withCondition('Ctr < ' + minCtr).
      withCondition('AverageCpc > ' + maxCpc).
      withCondition(labelCondition).
      forDateRange('LAST\_7\_DAYS').
      get();
  while (keywords.hasNext()) {
    var keyword = keywords.next();
    // We should mark it for review depending on whether it's a branded keyword or not.
    var label = isBrand(keyword.getText()) ? 'brand-review' : 'non-brand-review';
    // Check if it already has the label.
    if (!hasLabel(keyword, label)) {
      keyword.applyLabel(label);
      // Track it as newly labeled.
      newlyLabeled.push(keyword);
    }
  }
  return newlyLabeled
}

function labelBrandedKeywords() {
  var keywords = AdWordsApp.keywords().get();
  while (keywords.hasNext()) {
    var keyword = keywords.next();
    if (isBrand(keyword.getText())) {
      keyword.applyLabel('brand-keyword');
    }
  }
}

// Needs to be run (only once) before working with this script.
function createLabels() {
  AdWordsApp.createLabel('brand-review', 'Branded keywords needing review', 'red');
  AdWordsApp.createLabel('non-brand-review', 'Branded keywords needing review', 'maroon');
  AdWordsApp.createLabel('brand-keyword', 'Keywords that are part of our brand', 'blue');
}

/\*\*
 \* Returns true if this string is consider to be a part of our brand, false otherwise.
 \*/
function isBrand(s) {
  if (!s) {
    return false;
  }
  for (var i = 0; i < BRAND\_NAMES.length; i++) {
    if (s.toLowerCase().indexOf(BRAND\_NAMES[i].toLowerCase()) != -1) {
      return true;
    }
  }
  return false;
}

/\*\*
 \* Returns true if this keyword already has this label applied.
 \*/
function hasLabel(keyword, label) {
  return keyword.labels().withCondition("Name = '" + label + "'").get().hasNext();
}

/\*\*
 \* Generate a message body for all keywords that have been added to a label for review.
 \*/
function generateEmail(keywords) {
  var lines = [];
  for(var i = 0; i < keywords.length; i++) {
    lines.push([keywords[i].getCampaign().getName(), keywords[i].getAdGroup().getName(), keywords[i].getText()].join(' > '));
  }
  return lines.join('\n');
}